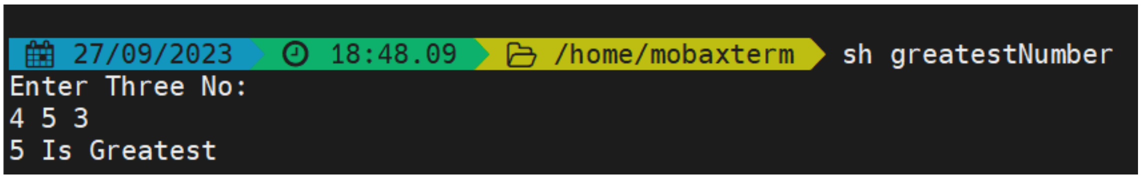
Journal programs:

1. Write a shell script to find greatest amongst three numbers.

Code:

```
echo "Enter Three No:"
read num1 num2 num3
if [ $num1 -gt $num2 ]; then
    if test $num1 -gt $num3
    then
        echo "$num1 Is Greatest"
    else
        echo "$num3 Is Greatest"
    fi
elif [ $num2 -gt $num3 ]; then
    echo "$num2 Is Greatest"
else
    echo "$num3 is greatest"
fi
```

Output:

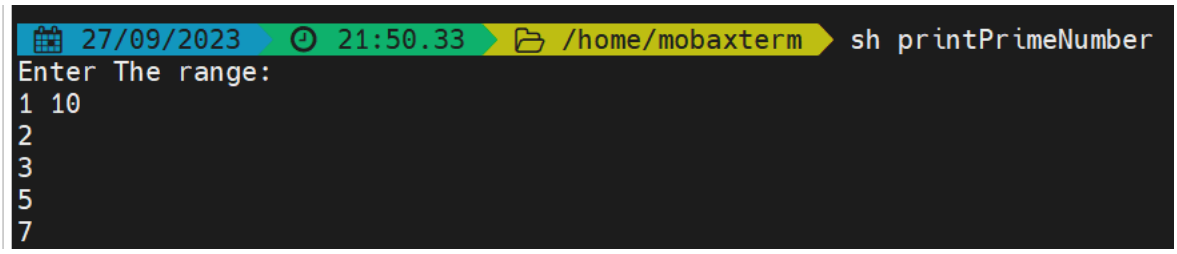


```
 27/09/2023   18:48.09   /home/mobaxterm   sh greatestNumber
Enter Three No:
4 5 3
5 Is Greatest
```

2. Write a shell script to find all prime numbers in given range.

```
echo "Enter The range: "
read start end
for(( i=$start; i<=$end; i++ ))
do
     check=0
     for a in $(seq 2 $( (expr $i / 2) ) )
```

```
        do
                if [ $( ( expr $i % $a ) ) -eq 0 ]
                then
                        check=1
                        break
                fi
        done
        if [ $check -eq 0 -a $i -ne 1 ]
        then
                echo $i
        fi
done
```



```
 27/09/2023   21:50.33   /home/mobaxterm   sh printPrimeNumber
Enter The range:
1 10
2
3
5
7
```

3. Write a shell script to draw following pattern.

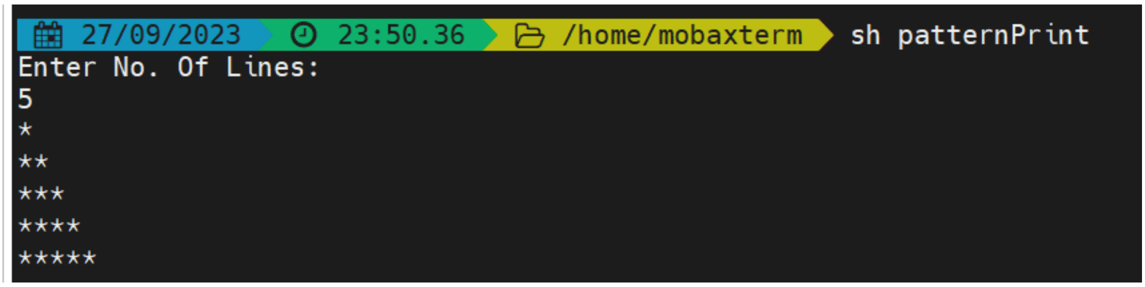
```
*
* *
* * *
* * * *
* * * * *
```

Code:’

```
echo "Enter No. Of Lines: "
read num
for ((i=1; i<=num; i++))
do
        for ((j=1; j<=i; j++))
        do
```

```
                echo -n "*"
        done
        echo " "
done
```

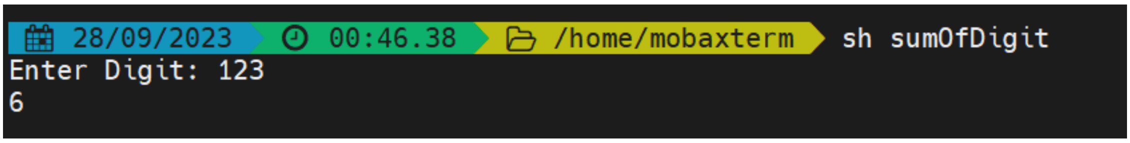
Output:



```
27/09/2023  23:50.36  /home/mobaxterm  sh patternPrint
Enter No. Of Lines:
5
*
**
***
****
*****
```

4. Write a shell script to find sum of digits of a number.

```
echo -n "Enter Digit: "
read num
sum=0
while [ $num -ne 0 ]
do
        digit=$(($num % 10))
        sum=$(($sum + $digit))
        num=$(($num / 10))
done
echo $sum
```



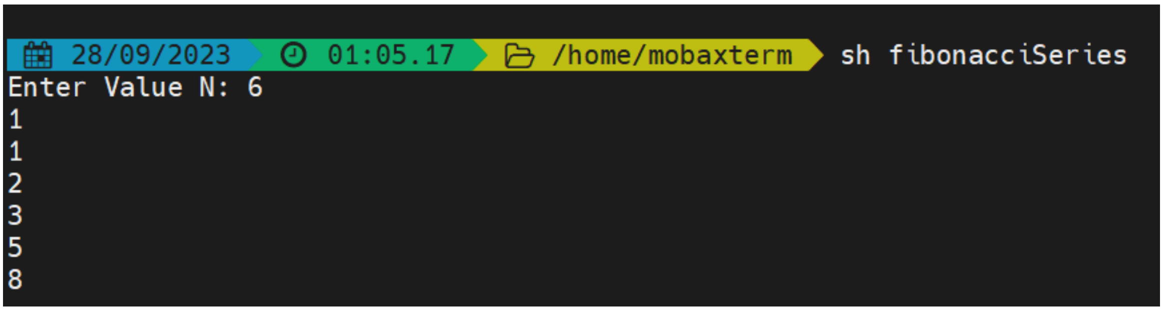
```
28/09/2023  00:46.38  /home/mobaxterm  sh sumOfDigit
Enter Digit: 123
6
```

5. Write a shell script to print fibonacci series upto entered value N.

Code:

```
echo -n "Enter Value N: "
read num
a=0
b=1
for i in $(seq 1 $(($num)))
```

```
do
        echo $b
        sum=$(( $a + $b))
        a=$b
        b=$sum
done
```



```
 28/09/2023   01:05.17   /home/mobaxterm   sh fibonacciSeries
Enter Value N: 6
1
1
2
3
5
8
```

6. Write a menu driven shell script which accepts basic amount as an input and displays following
options.
a. Dearness allowance (90% of basic)
b. Provident Fund F (12% of basic)
c. House Rent Allowance (20% of basic + DA)
d. Income tax deducted (5% of basic + DA + HRA)
e. Take home salary (basic + DA + HRA – IT)

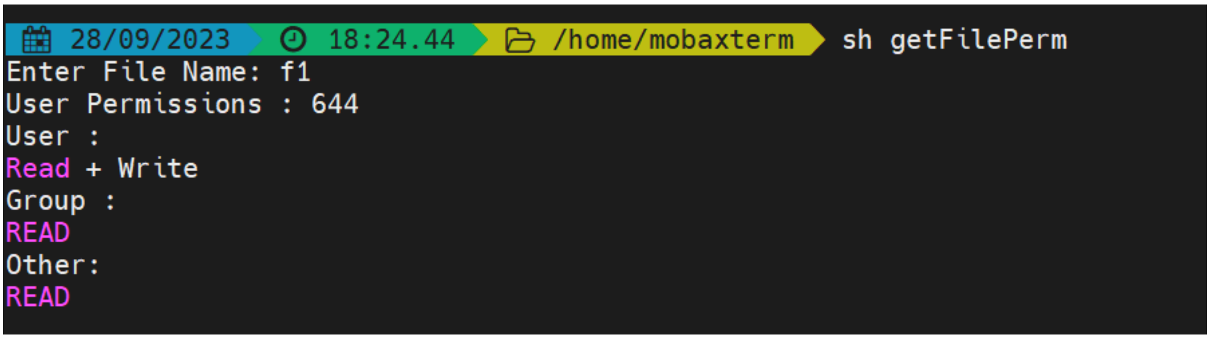
```
echo -n "Enter Basic Amount: "
read amount
echo ""
status=1
while [ $status -eq 1 ]
do
      echo "Options:"
      echo "1. Dearance Allowance"
      echo "2.Providant Fund"
      echo "3.House Rent Allowance"
      echo "4.Income tax deducted"
      echo "5.Take Home Salary"
      echo -n "Chhose Your Option: "
```

```
    read choice
    case $choice in
        1)
        echo "Dearance Allowance $((( $amount * 9 ) / 10))"
        ;;
        2)
        echo "Provident Fund $((( $amount * 12) / 100))"
        ;;
        3)
        echo "House Rent Allowance $((( $amount * 11) / 10))"
        ;;
        4)
        echo "Income Tax Deducted $((( $amount * 115) / 100))"
        ;;
        5)
        echo "Take Home Salary $((( $amount * 95) / 100))"
        ;;
        *)
        echo "Invalid Choice"
        ;;
esac
    echo -n "To Repeat Press 1 Or To Exit Press 0"
    read status
done

echo -n "Enter File Name: "
read file
if test -e $file
then

    perm=$(stat -c "%a" $file)
```

```
echo "User Permissions : $perm "
for((i=2; i >=0; i--))
do
        divisor=1
        for((j=1; j<=i; j++))
        do
                divisor=$(( $divisor * 10))
        done
        case $i in
                2)echo "User :"
                ;;
                1)echo "Group : "
                ;;
                0)echo "Other: "
                ;;
                *)echo "Not valid"
                ;;
        esac
        uPerm=$(( $perm / $divisor ))
        case $uPerm in
                0)echo -ne "No Permission\n"
                ;;
                1)echo -ne"execute permission\n"
                ;;
                2)echo -ne "write permission\n"
                ;;
                3)echo -ne "EXECUTE + WRITE\n"
                ;;
                4)echo -ne "READ\n"
                ;;
                5)echo -ne "Read + execute\n"
```

```
            ;;
            6)echo -ne "Read + Write\n"
            ;;
            7)echo -ne "Read +Write + Execute\n"
            ;;
            *)echo -ne "None\n"
            ;;
        esac
        perm=$(( $perm % $divisor))
    done

else
    echo "File does not exists"
fi
```

Output:



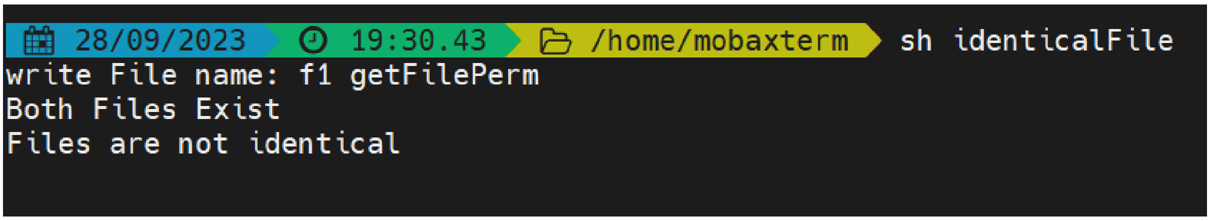
```
 28/09/2023   18:24.44   /home/mobaxterm   sh getFilePerm
Enter File Name: f1
User Permissions : 644
User :
Read + Write
Group :
READ
Other:
READ
```

8. Write a shell script that accepts two files are identical or not.

```
echo -e "write File name: \c"
read f1 f2
if [ ! -e $f1 ]
then
    echo " File 1 does not exist"
elif [ ! -e $f2 ]
then
```

```
        echo "File 2 does not exist"
else
        echo "Both Files Exist"
fi
cmp -s $f1 $f2
result=$?
case $result in
        0)echo "Files are identical"
        ;;
        1)echo "Files are not identical"
        ;;
        2)echo "Error"
        ;;
Esac
```

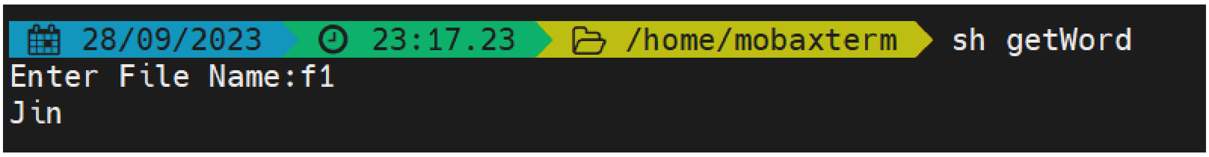
Output:



```
 28/09/2023   19:30.43   /home/mobaxterm  sh identicalFile
write File name: f1 getFilePerm
Both Files Exist
Files are not identical
```

9. Write a shell script to display all the words, having length <4 characters, of a file f1.txt

```
echo -e "Enter File Name:\c"
read file
echo `grep -oE '\b\w{1,3}\b' $file`
```

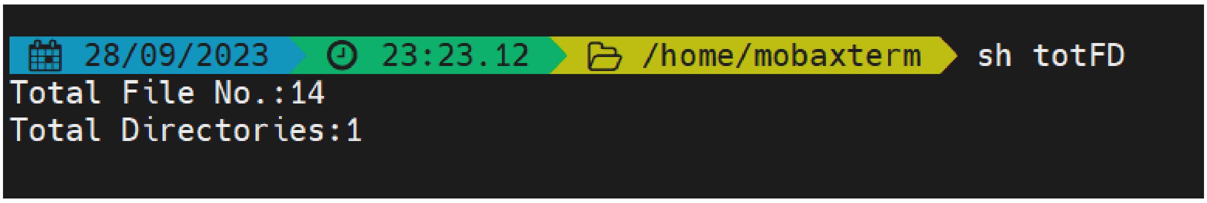


```
 28/09/2023   23:17.23   /home/mobaxterm  sh getWord
Enter File Name:f1
Jin
```

10. Write a shell script to find total number of files and total number of directories in current working
directory.

```
echo -e "Total File No.:\c "
echo | find . -maxdepth 1 -type f | wc -l
echo -e "Total Directories:\c"

echo | find . -maxdepth 1 -type d | wc –l
```

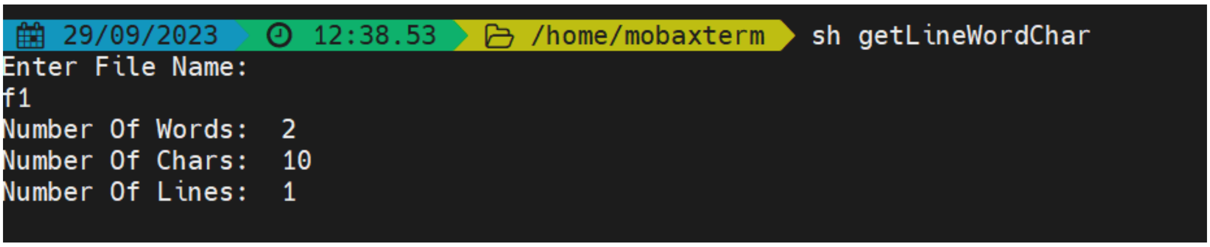
Output



```
 28/09/2023   23:23.12   /home/mobaxterm  sh totFD
Total File No.:14
Total Directories:1
```

11. Write a shell script to find total number of characters, words and lines of a file. (Do not use *wc*
command.

```
echo "Enter File Name: "

read file

awk '{

      chars+=length($0);

      words+=NF;

      lines++;

}

END{

      print "Number Of Words: ", words

      print "Number Of Chars: ", chars

      print "Number Of Lines: ", NR

}' $file
```



```
 29/09/2023   12:38.53   /home/mobaxterm  sh getLineWordChar
Enter File Name:
f1
Number Of Words:  2
Number Of Chars:  10
Number Of Lines:  1
```

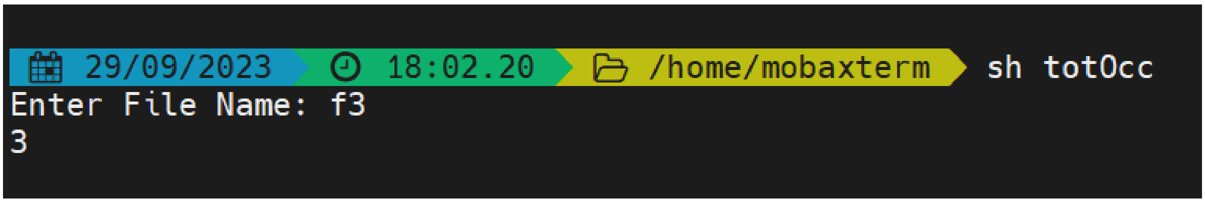
12. Write a shell script which accepts a username and check the entered user is currently logged in or
not.

```
#!/bin/bash
```

```
# Check if the user provided an argument
if [ $# -eq 0 ]; then
  echo "You need to enter a username"
  exit 1
fi

# Assign the argument to a variable
user=$1

# Use the who command to get the list of logged in users
# Use grep to search for the user name in the list
# Use wc to count the number of matches
result=$(who | grep $user | wc -l)

# If the result is greater than zero, the user is logged in
if [ $result -gt 0 ]; then
  echo "$user is currently logged in"
  exit 0
else
  echo "$user is not logged in"
  exit 1
fi
```

13. Write a shell script to find total number of occurrences of SDJIC in given file. (Please provide necessary validations)

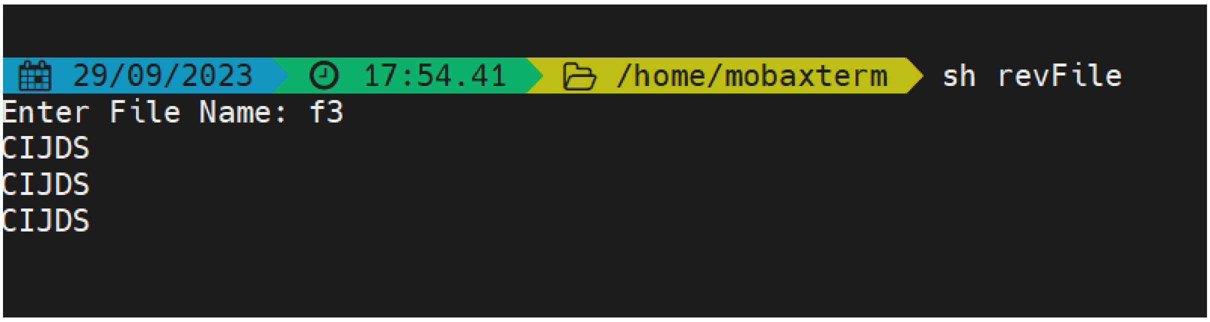
```
echo -n "Enter File Name: "
read file

if [ ! -f $file -o ! -r $file ]
then
        echo "File does not exist or is not readable"
        exit 1
fi
result=$( grep -ow 'SDJIC' $file | wc -l)
echo $result
```



```
29/09/2023  18:02.20  /home/mobaxterm  sh totOcc
Enter File Name: f3
3
```

14. Write a shell script which accepts filename as input and reverse individual words from it. (Please
provide necessary validations)

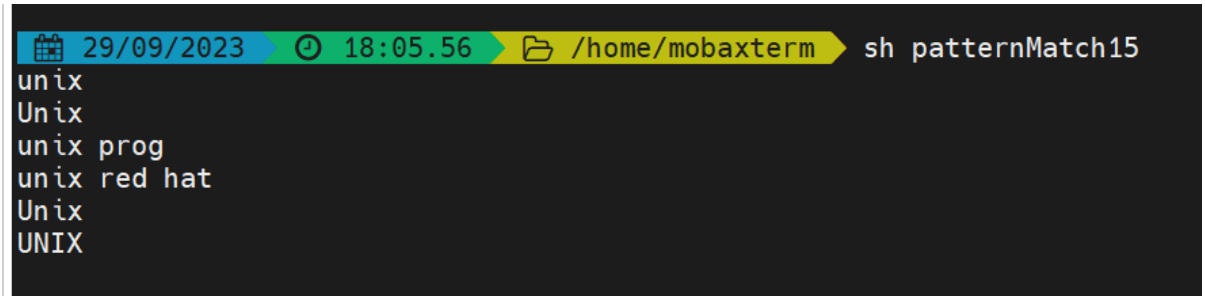
```
echo -n "Enter File Name: "
read file
if [ ! -f $file -o ! -r $file ]
then
    echo "File does not exist"
fi
awk '{
    for(i=1; i<=NF; i++){
        len = length($i)
        rev=""
        for(j=len; j>=1; j--){
            rev=rev substr($i, j, 1)
        }
        print rev
    }
    print ""
}' $file
```



```
 29/09/2023   17:54.41   /home/mobaxterm   sh revFile
Enter File Name: f3
CIJDS
CIJDS
CIJDS
```

15. Write a shell script to display all the lines from a file (11.txt), which starts with text “unix”. (not case
sensitive)

```
file=11

grep -i '^unix' $file
```



```
 29/09/2023   18:05.56   /home/mobaxterm   sh patternMatch15
unix
Unix
unix prog
unix red hat
Unix
UNIX
```

16. Write *grep* command to perform following actions:

a. Count number of blank lines in file f1.txt

   grep -c '^$' f1

b. print all lines containing *sdjic*

   grep ‘sdjic’ f1

c. print the lines that starts with *sdjic*.

   grep ‘^sdjic’ f1

d. Search the files in CPROGRAMS directory which has the string "include"

   grep -r ‘include’ CPROGRAMS
   grep -r '^echo' *

e. print lines having exactly 50 characters in file f1.txt

```
grep ‘^.\{50\}$’ f1
```

f. Count number of blank lines in file f1.txt

```
grep -c '^$' f1
```

g. Display lines having atleast one characters in file f1.txt

```
grep  ‘.\{1,\}’ f1
```

h. Display lines having *sdjic* text in any case in file f1.txt

```
grep –I ‘sdjic’ f1
```

i. Display line of file f1.txt having exactly 3 characters

```
grep ‘^.\{3\}$’ f1
```

j. Display lines of file f1.txt which begin with any alphabet

```
grep ‘^[a-zA-Z]’ f1
```

k. Display lines whose last word is “UNIX” in file f1.txt

```
grep ‘UNIX$’ f1
```

l. Display filenames having last character as digit [0-9]

m. Display list of filenames that only consist digits

n. Display line of file f1.txt which only consist digits

```
grep ‘^[0-9]*$’ f1
```

o. Display lines of file f1.txt which only consist capital alphabets

```
grep ‘^[A-Z]*$’ f1
```

p. Search all lines in file f1.txt which ends with “.”

```
grep ‘\.$’ f1
```

17. Write sed command to perform following tasks

a. To print only last line of f1.txt

```
sed –n ‘$p’ f1
```

b. To print line number 1-3, 6-7 and 10 of f1.txt

```
sed –ne ‘1,3p’ –e ‘6,7p’ –e ‘10p’ f1
```

c. To print lines beginning with SDJIC of f1.txt

```
sed –n ‘/^unix/p’ f1
```

d. Print three lines starting from fourth line of f1.txt

```
sed –n ‘4,6p’ f1
```

e. Print all blank lines of file f1.txt

```
sed –n ‘/^$/p’ 11
```

f. Print lines having either of “sdjic” or “sdjyc”

```
sed –ne ‘/sdj[iy]c/p’ f1
```

g. Lines beginning with either alphabet or digit

```
sed –n ‘/^[0-9A-Za-z]/p’ f1
```

h. To insert a line “additional line” before every line

```
sed ‘i\
> additional line
> f1
```

i. To replace every occurrence of | with : of first three lines

```
sed ‘1,3s/|/:/g’ f1
```

j. To replace every occurrence of “|” with “:” of every line

```
sed ‘s/|/:/g’ f1
```

k. To remove all the lines having word “fail” from file f1.txt (delete command)

```
sed ‘/fail/d’ f1
```