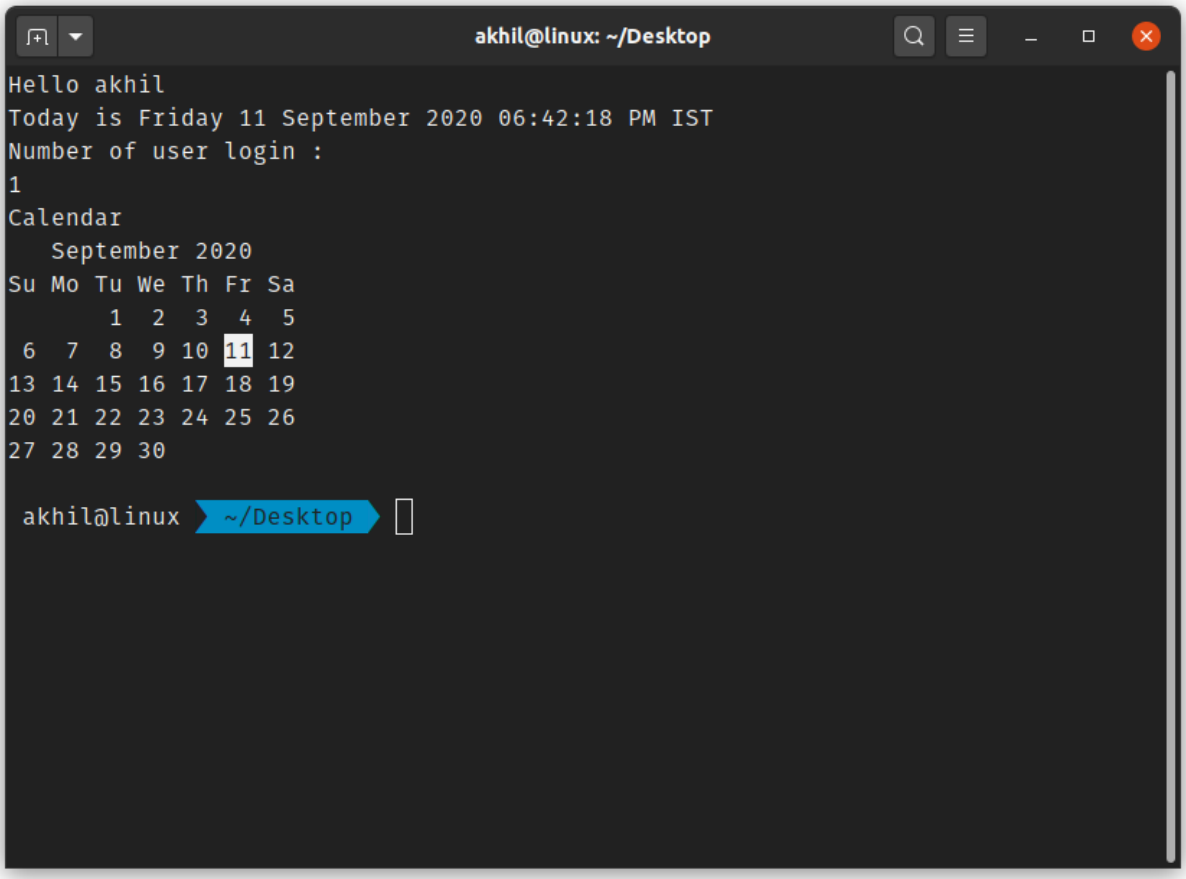
Exercise 1:

1)Write the following shell script, save it, execute it and note down the output.

```
# Script to print user information who currently login, current date & time
# Enter the following commands in a file
#
clear
echo "Hello $USER"
echo "Today is ";date
echo "Number of user login : " ; who | wc -l
echo "Calendar"
cal
exit 0
```

Output



Exercise 2:

1) If you want to print your home directory location then you give command:
a)echo $HOME
OR
(b)echo HOME
Which of the above command is correct & why?
Caution: Do not modify System variable, this can some time create problems.

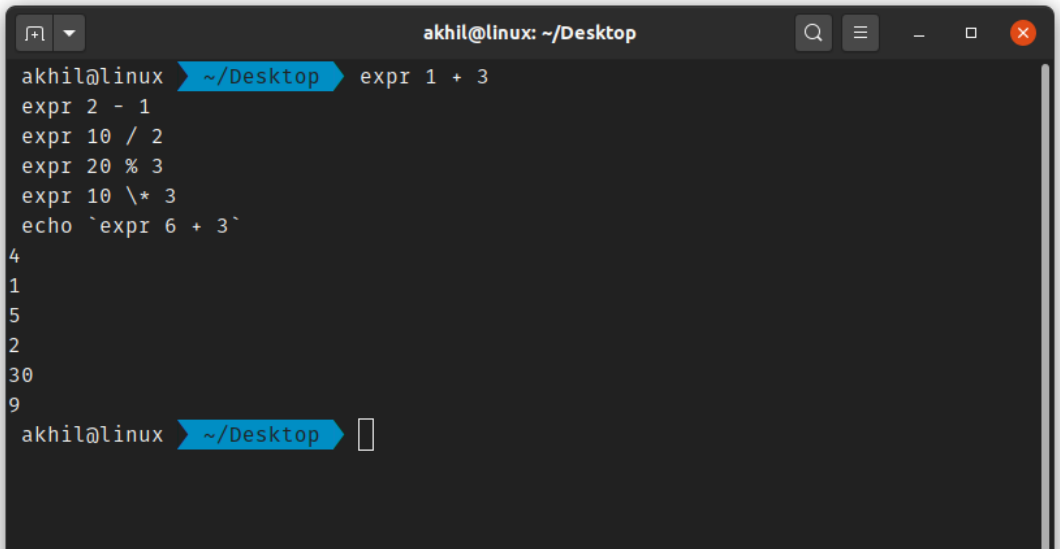
ANS - $HOME as # is used to print values held by variables in shell scripting . And HOME is an environment variable in linux that prints the path to the home directory of the user.

Exercise 3:

What is the output of the following expressions?

$ expr 1 + 3
$ expr 2 - 1
$ expr 10 / 2
$ expr 20 % 3
$ expr 10 * 3
$ echo `expr 6 + 3`

Output



Exercise 4:

What is the meaning of Single quote (‘), Double quote (“) and Back quote (`) in shell?

Ans -

Enclosing characters in single quotes **(')** preserves the literal value of each character within the quotes. A single quote cannot be escaped between single quotes, even when preceded by a backslash.

Enclosing characters in double quotes (") preserves the literal value of all characters within the quotes, with the exception of $, `, \, and, when history expansion is enabled, !. The characters $ and ` retain their special meaning within double quotes .The backslash retains its special meaning only when followed by one of the following characters: $, `, ", \, or newline. Within double quotes, backslashes that are followed by one of these characters are removed(as the character following it escaped). Backslashes preceding characters without a special meaning are left unmodified. A double quote may be quoted within double quotes by preceding it with a backslash.

```
#!/bin/sh
MYVAR="testString"
echo "Usage of ' and \" : "
echo $MYVAR
echo "double quotes gives you $MYVAR"
echo 'single quotes gives you $MYVAR'
CMD='ls'
echo "Usage of \` :"
echo `$CMD`
```

Output

```
akhil@linux: ~/Desktop
 akhil@linux  ~/Desktop  bash second.sh
Usage of ' and " :
testString
double quotes gives you testString
single quotes gives you $MYVAR
Usage of ` :
first.sh hello hello.cpp second.sh
 akhil@linux  ~/Desktop
```

Exercise 5:

What does the following command do?

$ sort <myfile>sorted_file

Ans - sort <myfile>sorted_file command sorts the file line by line and using -o(output) argument we can put the contents of this file into another file as shown below

```
✘ akhil@linux  ~/Desktop  sort unsorted.txt -o sorted.txt
 akhil@linux  ~/Desktop  cat sorted.txt
Apple
Bell
Boy
Cat
Dog
Train
Zebra
 akhil@linux  ~/Desktop
```

Exercise 6:

Create a shell script (using Bourne Shell or Bash) which converts all file names in the current directory to lowercase. You should execute the script and send a screenshot of the output.

Example:

If you have these files in the current directory: ABC, foo1.c, Test, sample, foo2.TXT, myFile.Xa

Expected output after executing the script: abc, foo1.c, test, sample, foo2.txt, myfile.xa

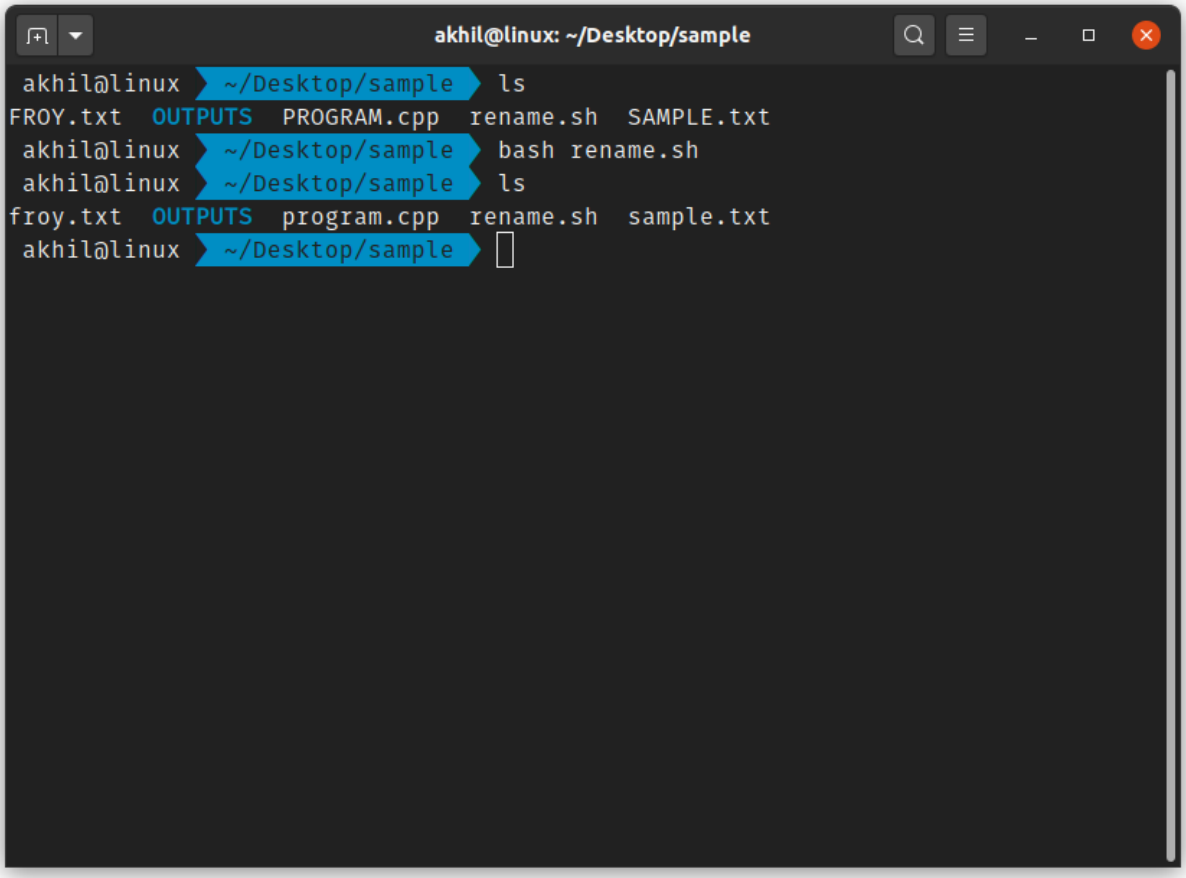
CODE:

```
#!/bin/sh
#
for item in `ls`
     do
       if [ ! -f $item ]; # the ! -f $item is file
                          # checker which checks
                          #if file is a plain file
       then
          continue #(rejects al directories)

          fi #end of if-else

        new=`echo $item  | tr '[A-Z]' '[a-z]'` #new filename renamed
                                               #using output of echo to tr
                                               # which changes uppercase to
                                               #lowercase based on flags
        if [ $new != $item ]; then  #if name has been changed
          mv -i $item $new #rename and  -i to ask permission if overwrite
        fi #end of if-else
        done
```

OUTPUT:


akhil@linux: ~/Desktop/sample
akhil@linux ~/Desktop/sample ls
FROY.txt OUTPUTS PROGRAM.cpp rename.sh SAMPLE.txt
akhil@linux ~/Desktop/sample bash rename.sh
akhil@linux ~/Desktop/sample ls
froy.txt OUTPUTS program.cpp rename.sh sample.txt
akhil@linux ~/Desktop/sample