Prac_1 Check the output of the following commands:

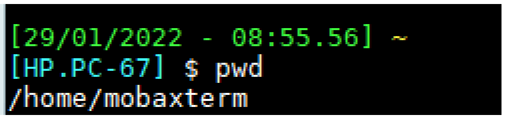
date, ls, who, cal, ps, wc, cat, uname, pwd, mkdir, rmdir, cd, cp, rm, mv, diff, chmod, grep, sed, head, tail,cut, paste, sort, find, man

Prac_2 Write a script to find the complete path for any file.

```
clear
echo "Enter File Name : \c "
read fileName

if [ -f $fileName ]
then
str=`find $fileName`
path=`pwd`
echo "Full path of file is $path/$str"

else
echo "file [ $fileName ] not exist in \c "
pwd
fi
```

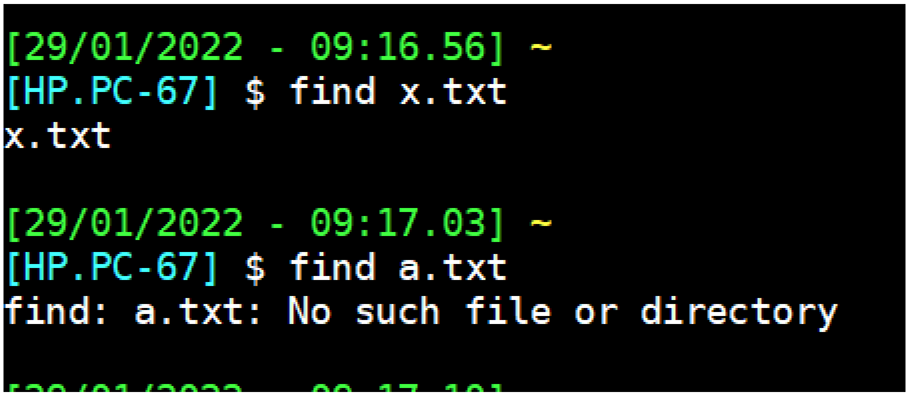
What is the pwd command example?

Simply type pwd into your terminal, and the command will output the absolute path of your **present working directory**. In this example, the command has indicated that our present working directory is



```
[29/01/2022 - 08:55.56] ~
[HP.PC-67] $ pwd
/home/mobaxterm
```

Find
Find command is use for find the file



```
[29/01/2022 - 09:16.56] ~
[HP.PC-67] $ find x.txt
x.txt

[29/01/2022 - 09:17.03] ~
[HP.PC-67] $ find a.txt
find: a.txt: No such file or directory
```

Prac_3 Write a shell script to execute following command

```
echo sorting the file

sort abc.txt > xyz.txt

echo executing two commands

who ; ls

echo -e " this is \n a three-line \n Text message "# use -e option if required

echo The version is `uname -a`

echo Help of cat command
```

note:

in above program create two file abc.txt xyz.txt

using cat > abc.txt

cat > xyz.txt

save with ctr+z

Prac_4 Write a shell script to execute following command

```
echo 1. How would u display the hidden files
echo 2. How delete directory with files
echo 3. How would user can do interactive copying
echo 4. How would user can do interactive deletion of files
echo 5. Explain two functionalities of mv command with example
echo enter your choice
read ch
case $ch in
1) echo Displaying hidden files
ls .[a-z]* ;;
2) echo Deleting directories with files
rm -R abc.txt ;;
3) echo Interactive copy
cp -i a.txt xyz.txt ;; # file2 should be created first to check interactivity
4) echo Interactive Deletion
rm -i x.txt ;;
5) echo mv command
mv xyz.txt x.txt ;;
*) echo Invalid choice ;;
esac
```

5 Write a shell script to execute following commands

1. Create a file called text and store name, age and address in it.

2. Display the contents of the file text on the screen.

3. Delete the directories mydir and newdir at one shot.

4. Sort a numeric file?

5. Change the permissions for the file newtext to 666.

```
echo “1. Create a file called text and store name, age and address in it.”

echo “2. Display the contents of the file text on the screen.”

echo “3. Delete the directories mydir and newdir at one shot.”

echo “4. Sort a numeric file”

echo “5. Change the permissions for the file newtext to 666.”

echo “enter your choice”

read ch

case $ch in

1) echo “Create a file called text and store name,age and address in it.”

echo “Enter the filename”

read fn

cat > $fn  ;;

2) echo “Display the contents of the file text on the screen.”

cat $fn ;;

3) echo “Delete the directories mydir and newdir at one shot.”

rmdir mydir newdir  ;;

4)  echo “Sort a numeric file”

     sort -n filename ;;
```

```
5) echo “Change the permissions for the file newtext to 666.”

   chmod 666 newtext ;;

*) echo “Invalid choice” ;;

esac
```

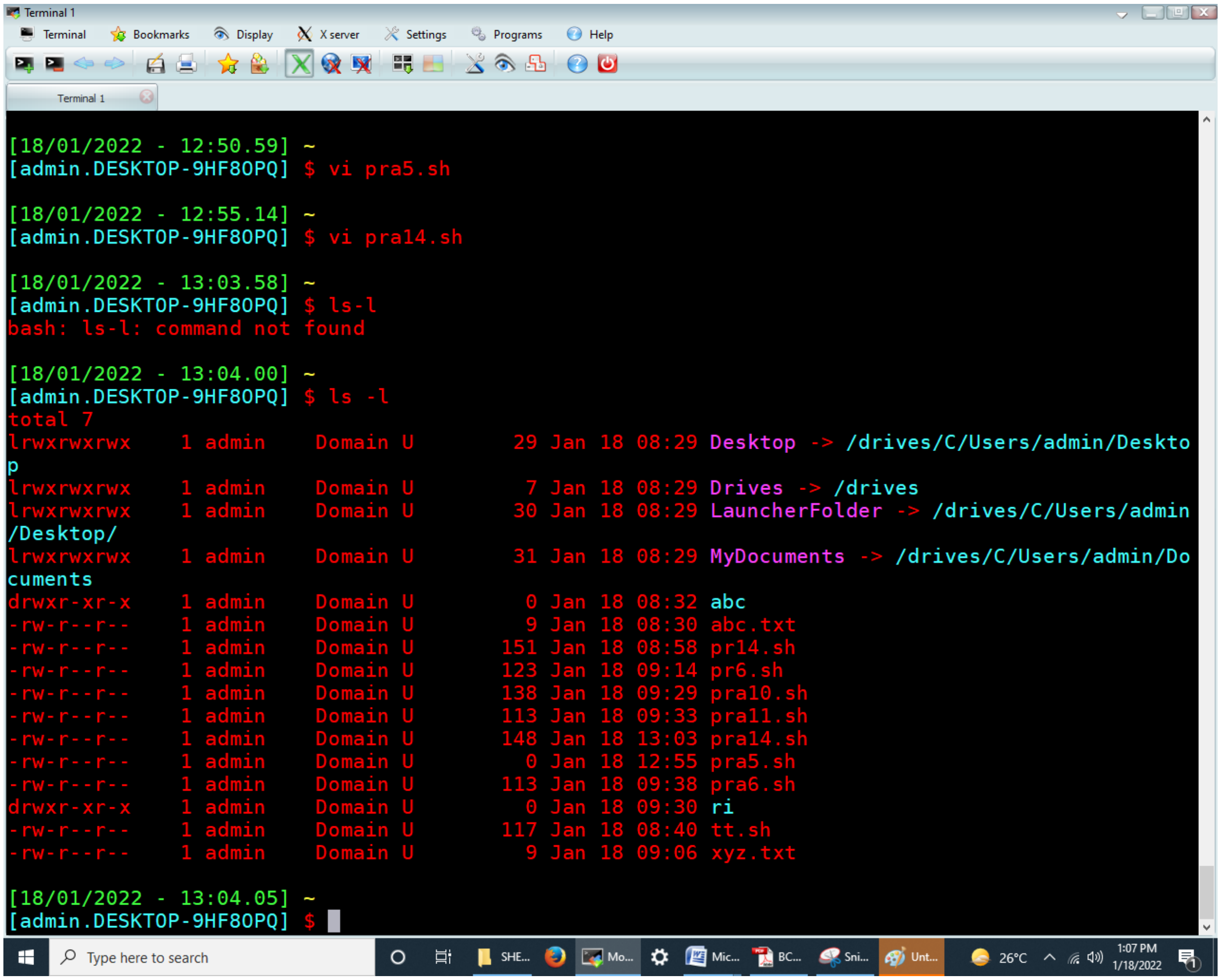
Pra-6: Write shell script that accept filename and displays last modification time if file exists.

```
echo "enter filename"
read fn
if [ -e $fn ]
then
ls -l $fn | cut -d " " -f 23
else
echo "file does not exit"
fi
```

NOTE

```
ls -l $fn | cut -d " " -f 23
```

here ls-l display in detail list



```
[18/01/2022 - 12:50.59] ~
[admin.DESKTOP-9HF80PQ] $ vi pra5.sh

[18/01/2022 - 12:55.14] ~
[admin.DESKTOP-9HF80PQ] $ vi pra14.sh

[18/01/2022 - 13:03.58] ~
[admin.DESKTOP-9HF80PQ] $ ls-l
bash: ls-l: command not found

[18/01/2022 - 13:04.00] ~
[admin.DESKTOP-9HF80PQ] $ ls -l
total 7
lrwxrwxrwx    1 admin    Domain U       29 Jan 18 08:29 Desktop -> /drives/C/Users/admin/Deskto
p
lrwxrwxrwx    1 admin    Domain U        7 Jan 18 08:29 Drives -> /drives
lrwxrwxrwx    1 admin    Domain U       30 Jan 18 08:29 LauncherFolder -> /drives/C/Users/admin
/Desktop/
lrwxrwxrwx    1 admin    Domain U       31 Jan 18 08:29 MyDocuments -> /drives/C/Users/admin/Do
cuments
drwxr-xr-x    1 admin    Domain U        0 Jan 18 08:32 abc
-rw-r--r--    1 admin    Domain U        9 Jan 18 08:30 abc.txt
-rw-r--r--    1 admin    Domain U      151 Jan 18 08:58 pr14.sh
-rw-r--r--    1 admin    Domain U      123 Jan 18 09:14 pr6.sh
-rw-r--r--    1 admin    Domain U      138 Jan 18 09:29 pra10.sh
-rw-r--r--    1 admin    Domain U      113 Jan 18 09:33 pra11.sh
-rw-r--r--    1 admin    Domain U      148 Jan 18 13:03 pra14.sh
-rw-r--r--    1 admin    Domain U        0 Jan 18 12:55 pra5.sh
-rw-r--r--    1 admin    Domain U      113 Jan 18 09:38 pra6.sh
drwxr-xr-x    1 admin    Domain U        0 Jan 18 09:30 ri
-rw-r--r--    1 admin    Domain U      117 Jan 18 08:40 tt.sh
-rw-r--r--    1 admin    Domain U        9 Jan 18 09:06 xyz.txt

[18/01/2022 - 13:04.05] ~
[admin.DESKTOP-9HF80PQ] $
```

Cut -d [Delimiter]

Use for extract field from a list/file.

-f 23 use for particular field and which line you want to see.

Other example

Extracting field from file

Remember, in order to extract a field from a file, we would need a delimiter (i.e. a column separator), based on which the file will be divided into columns and we can extract any of them. In this case, the syntax would be-

```
cut -d [DELIMITER] -f [RANGE] [FILENAME]
```

Here, we are instructing `cut` command to use a particular delimiter with option `-d` and then extract certain fields using option `-f`.

1. Display a specific field from a file

In case of a *csv* file, it is crystal clear that our delimiter will be a comma (,). Now, we need to enlist the names of the employees working in our organization, i.e. field number 2.

Example:

```
$ cut -d ',' -f 2 employees.txt
Employee Name
John Davies
Mary Fernandes
Jacob Williams
Sean Anderson
Nick Jones
Diana Richardson
```

Looks good.

2. Displaying Multiple Fields from a File

Moving forward now, lets display more than one field now. Suppose, we need to include 'Age' and 'Gender' fields also. For this, we must specify the range - again, a start and an end.

```
$ cut -d ',' -f 2-4 employees.txt
Employee Name, Age, Gender
John Davies, 35, M
Mary Fernandes, 29, F
Jacob Williams, 40, M
Sean Anderson, 25, M
Nick Jones, 42, M
Diana Richardson, 29, F
```

7 Write a shell script to display the login names that begin with ‘s’.

```
who | grep ^s
```

8 Write a shell script to remove the zero sized file from the current directory

```
for i in *

do

if [ ! -s $i ]

then



rm $i

echo " $i removed "

fi

done
```

9 Write a shell script to display the name of all the executable file from the current directory.

```
countx=0
for i in *
 do
 if [ -x $i ]
 then
  countx=`expr $countx + 1`
   fi
  done
   echo Number of executable files are $countx
```

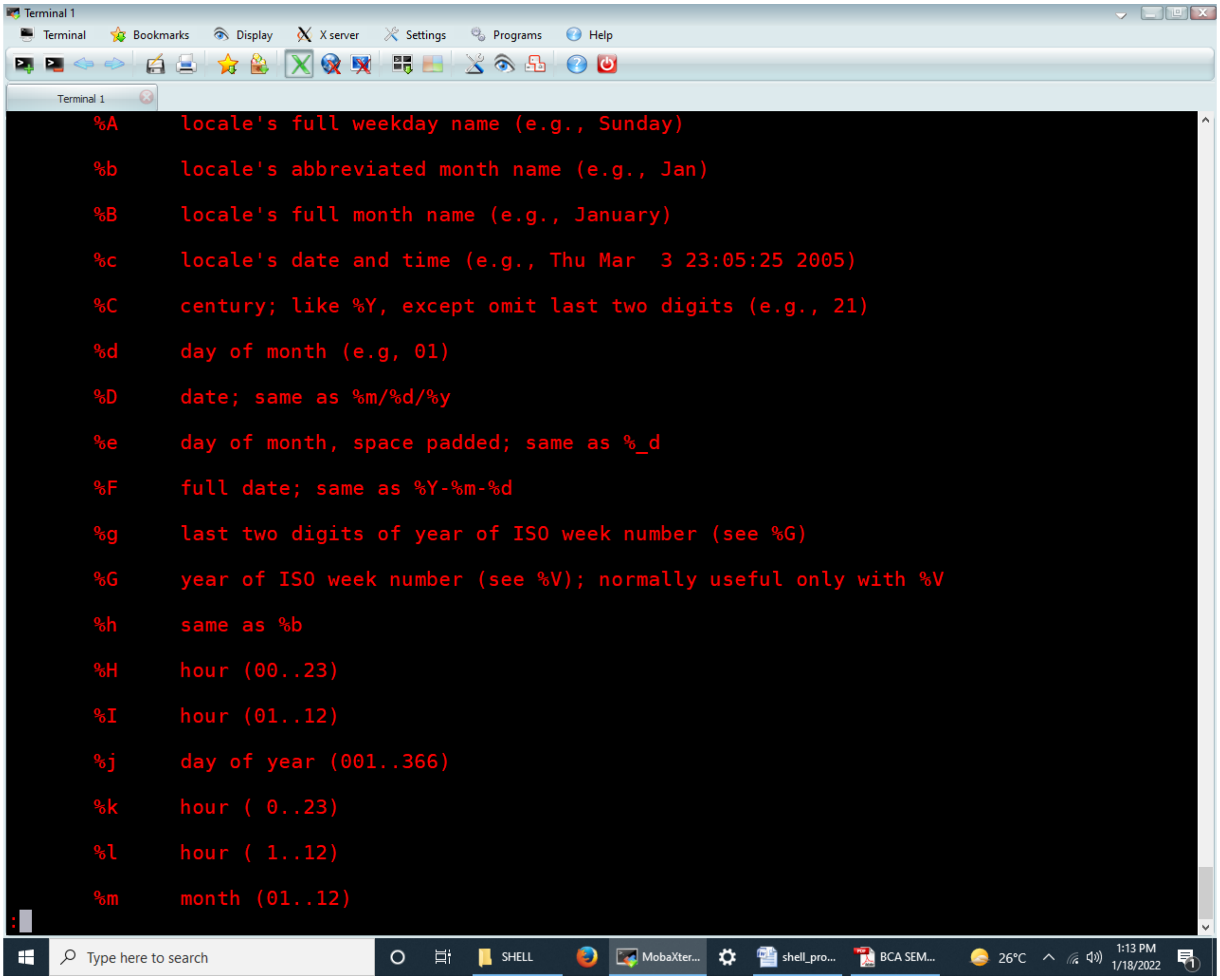
10. Write a shell script that will display welcome message according to time

```
d=`date +"%H"`
if [ $d -lt 12 ]
then
echo Good Morning
elif [ $d -gt 12 -a $d -lt 14 ]
then
echo good afternoon
else
echo good night
fi
```

Note

Here date command display date

```
[admin.DESKTOP-9HF8OPQ] $ date
Tue Jan 18 13:10:40 IST 2022

NAME
       date - print or set the system date and time

SYNOPSIS
       date [OPTION]... [+FORMAT]
       date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
```



```
       %A     locale's full weekday name (e.g., Sunday)

       %b     locale's abbreviated month name (e.g., Jan)

       %B     locale's full month name (e.g., January)

       %c     locale's date and time (e.g., Thu Mar  3 23:05:25 2005)

       %C     century; like %Y, except omit last two digits (e.g., 21)

       %d     day of month (e.g, 01)

       %D     date; same as %m/%d/%y

       %e     day of month, space padded; same as %_d

       %F     full date; same as %Y-%m-%d

       %g     last two digits of year of ISO week number (see %G)

       %G     year of ISO week number (see %V); normally useful only with %V

       %h     same as %b

       %H     hour (00..23)

       %I     hour (01..12)

       %j     day of year (001..366)

       %k     hour ( 0..23)

       %l     hour ( 1..12)

       %m     month (01..12)

:
```

11. Write a shell script to find number of ordinary files and directory files

```
countd=0
countf=0
for i in *
do
  if [ -d $i ]
  then
   countd=`expr $countd + 1`
   fi
    if [ -f $i ]
   then
  countf=`expr $countf + 1`
    fi
    done
    echo Number of directories are $countd
    echo Number of Ordinary files are $countf
```

12 Write a shell script that takes a filename from the command line and checks whether the file is an ordinary file or not.

- If it is an ordinary file then it should display the contents of the file.
- If it is not an ordinary file then script should display the message: “File does not exist or is not ordinary, cannot display.

```
echo "enter filename"
read $1
if [ -f $1 ]
then
echo Its an ordinary file
else
echo File does not exist or is not ordinary file
fi
```

13 Write a shell script that takes a filename from the user and checks whether it is a directory file or not.

```
echo enter the filename
read fn
if [ -f $fn ]
then
echo Its a directory ls $fn
else
echo Its not a directory
fi
```

Note

-f is used for file

-d used for directory

14 Write a shell script that takes a filename as an argument and checks if the file exists and is executable.

```
echo enter file name
read fn
if [ -e $fn -a -x $fn ]
then
echo file exists and is executable
else
echo file does not exist or is not executable
fi
```

Note

-e $fn -a -x $fn

-e used for exist

$fn used for file name

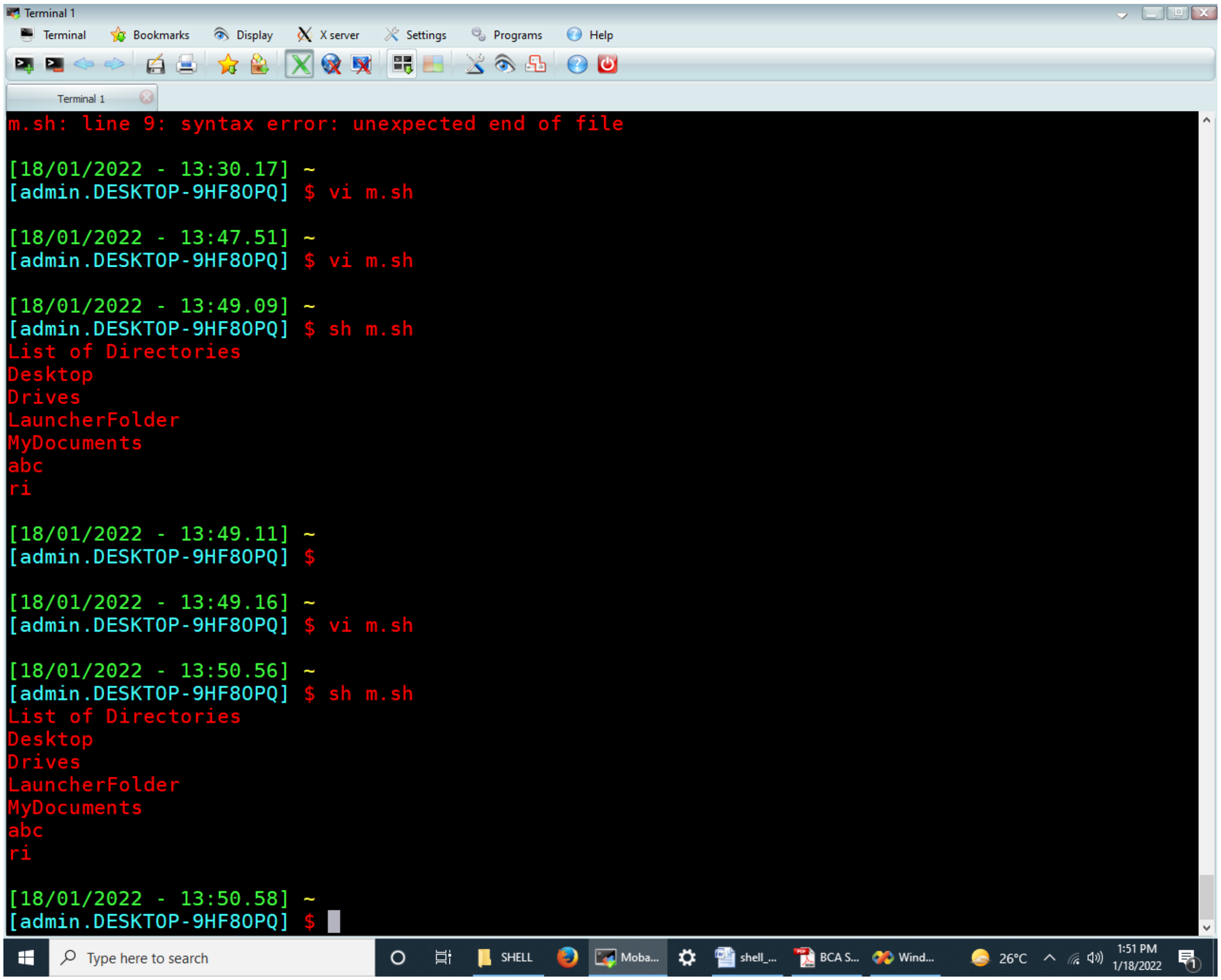
-a means and

-x means executable

15 Write a shell script that displays all subdirectories in current working directory.

```
echo List of Directories
for i in *
do
 if [ -d $i ]
 then
 echo $i
 fi
done
```

output



```
m.sh: line 9: syntax error: unexpected end of file

[18/01/2022 - 13:30.17] ~
[admin.DESKTOP-9HF80PQ] $ vi m.sh

[18/01/2022 - 13:47.51] ~
[admin.DESKTOP-9HF80PQ] $ vi m.sh

[18/01/2022 - 13:49.09] ~
[admin.DESKTOP-9HF80PQ] $ sh m.sh
List of Directories
Desktop
Drives
LauncherFolder
MyDocuments
abc
ri

[18/01/2022 - 13:49.11] ~
[admin.DESKTOP-9HF80PQ] $

[18/01/2022 - 13:49.16] ~
[admin.DESKTOP-9HF80PQ] $ vi m.sh

[18/01/2022 - 13:50.56] ~
[admin.DESKTOP-9HF80PQ] $ sh m.sh
List of Directories
Desktop
Drives
LauncherFolder
MyDocuments
abc
ri

[18/01/2022 - 13:50.58] ~
[admin.DESKTOP-9HF80PQ] $
```

16 Write a shell script that calculates the number of ordinary and directory files in your current working directory.

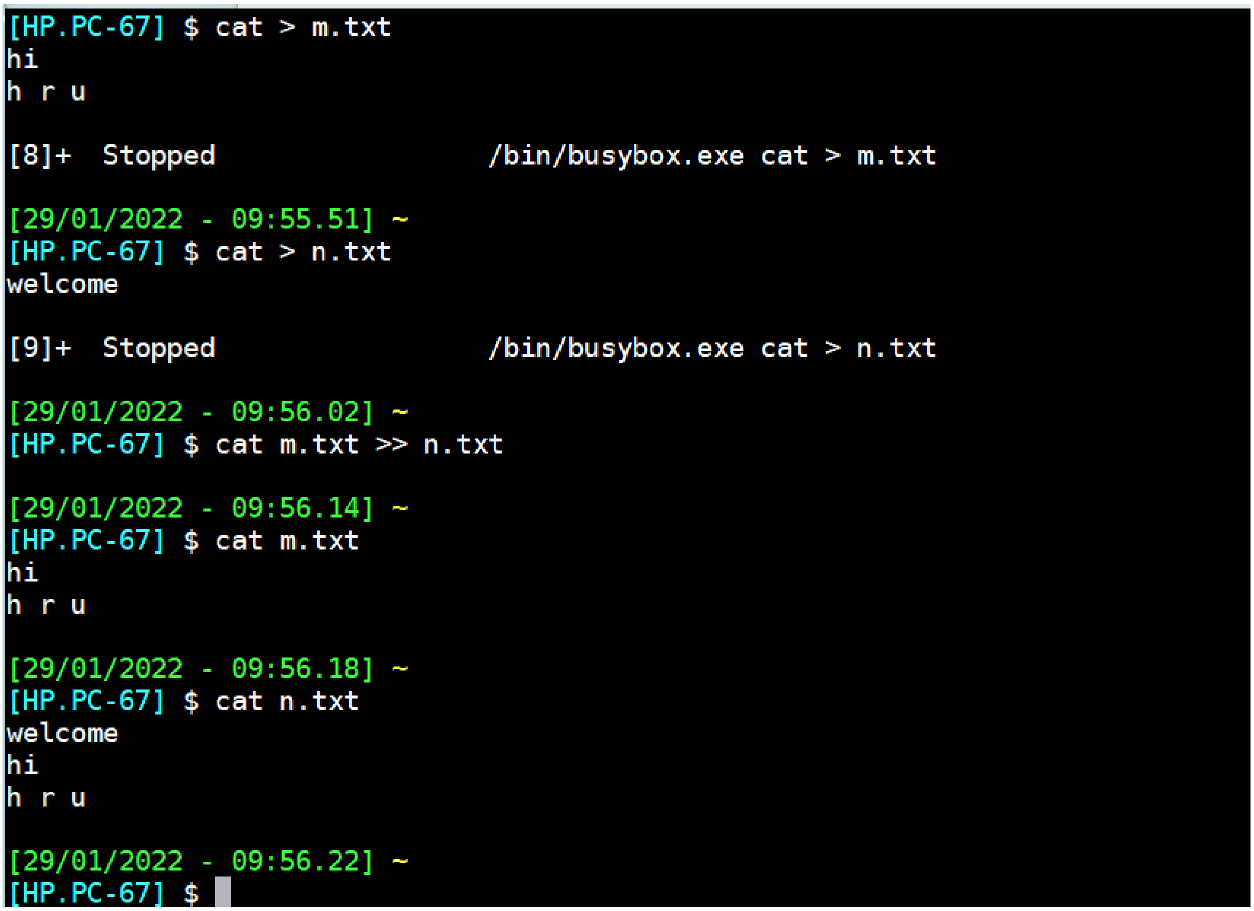
```
countd=0
countf=0
for i in *
do
  if [ -d $i ]
  then
   countd=`expr $countd + 1`
   fi
    if [ -f $i ]
   then
  countf=`expr $countf + 1`
    fi
    done
    echo Number of directories are $countd
    echo Number of Ordinary files are $countf
```

17 Write a shell script that accepts 2 filenames and checks if both exists; if both exist then append the content of the second file into the first file

```
echo enter the first filename
read fn1
echo enter the second filename
read fn2
if [ -f $fn1 -a -f $fn2 ]
then
echo Both file exists
cat $fn2 >> $fn1
else
echo Files does not exist
fi
```

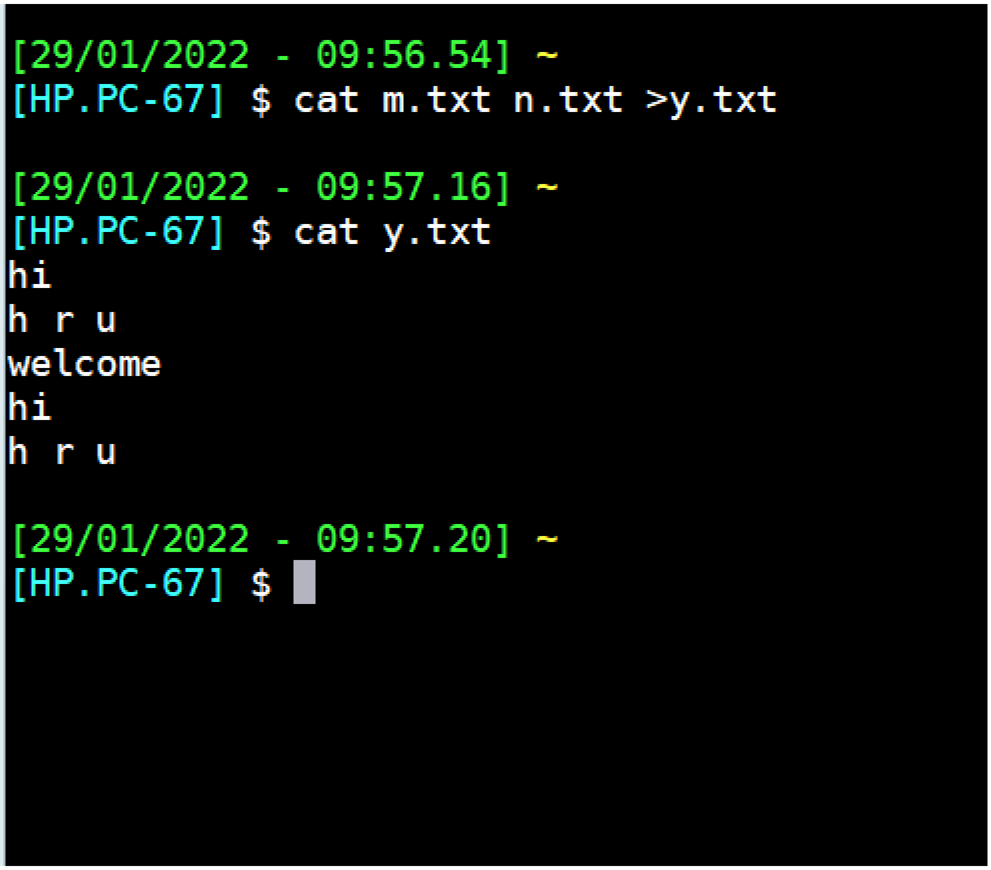
Note:

In above program create two file



```
[HP.PC-67] $ cat > m.txt
hi
h r u

[8]+  Stopped                 /bin/busybox.exe cat > m.txt

[29/01/2022 - 09:55.51] ~
[HP.PC-67] $ cat > n.txt
welcome

[9]+  Stopped                 /bin/busybox.exe cat > n.txt

[29/01/2022 - 09:56.02] ~
[HP.PC-67] $ cat m.txt >> n.txt

[29/01/2022 - 09:56.14] ~
[HP.PC-67] $ cat m.txt
hi
h r u

[29/01/2022 - 09:56.18] ~
[HP.PC-67] $ cat n.txt
welcome
hi
h r u

[29/01/2022 - 09:56.22] ~
[HP.PC-67] $
```

You can also contain copies to new file using follow command .in below command we merge two file and contain display new file.



```
[29/01/2022 - 09:56.54] ~
[HP.PC-67] $ cat m.txt n.txt >y.txt

[29/01/2022 - 09:57.16] ~
[HP.PC-67] $ cat y.txt
hi
h r u
welcome
hi
h r u

[29/01/2022 - 09:57.20] ~
[HP.PC-67] $
```

18 Write a shell script that takes the name of two files as arguments and performs the following:

i. Displays the message : “Displaying the contents of file :(first argument)” and displays the contents page wise.

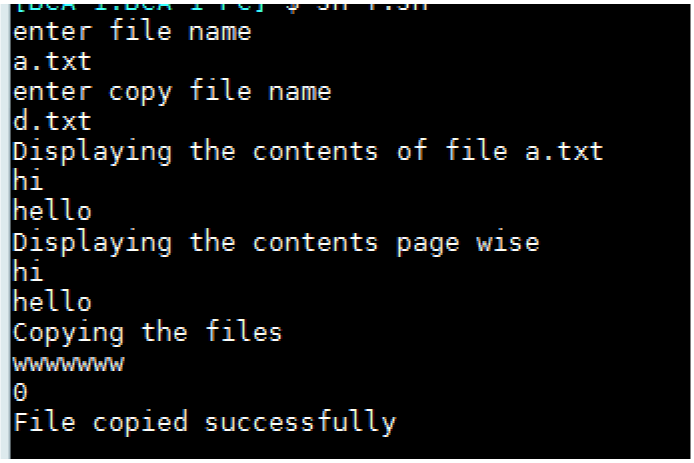
ii. ii. Copies the contents of the first argument to second argument.

iii. iii. Finally displays the message: “File copied successfully.”

```
echo enter file name
read n
echo enter copy file name
read file
echo Displaying the contents of file $n
cat $n
echo Displaying the contents page wise
cat $n | more
```

```
echo Copying the files
cp $n $file
c=`echo $?`
if [ $c -eq 0 ]
then
echo File copied successfully
else
echo Files not copied successfully
fi
```

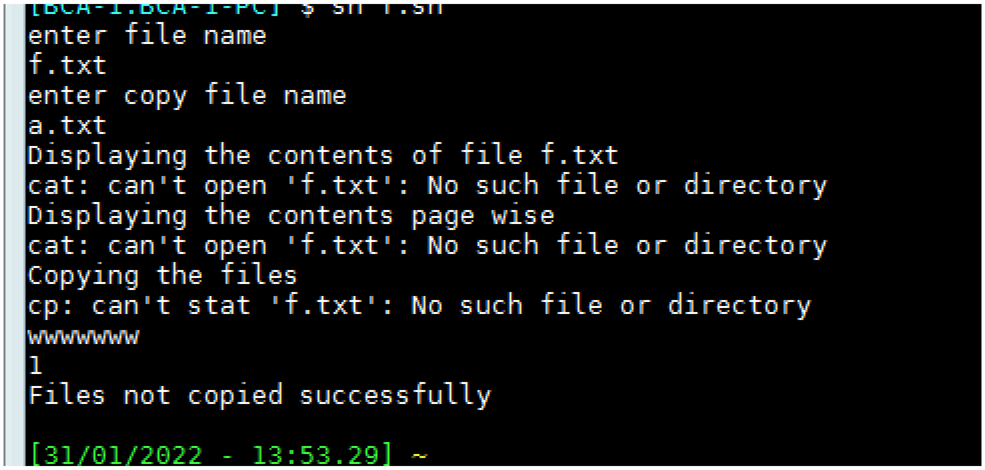
Note:



```
[BCA 1:BCA 1 r c] $ sh f.sh
enter file name
a.txt
enter copy file name
d.txt
Displaying the contents of file a.txt
hi
hello
Displaying the contents page wise
hi
hello
Copying the files
wwwwwww
0
File copied successfully
```

Echo $?

Echoes (prints) **the exit value for the previous command**. If it failed it will be different than zero (0). $ cd ~ $ echo $? > 0 $ cd //whatever/ > bash: cd: //whatever/: No such file or directory $ echo $? > 1. Programs exit with a status code.



```
[BCA-1.BCA-1-PC] $ sh f.sh
enter file name
f.txt
enter copy file name
a.txt
Displaying the contents of file f.txt
cat: can't open 'f.txt': No such file or directory
Displaying the contents page wise
cat: can't open 'f.txt': No such file or directory
Copying the files
cp: can't stat 'f.txt': No such file or directory
wwwwwww
1
Files not copied successfully

[31/01/2022 - 13:53.29] ~
```

19 Write a shell script to display the following menu and acts accordingly:

i. Calendar of the current month and year.

ii. Display “Good Morning/Good Afternoon/Good Evening” according tothe current login time.

iii. User name, Users home directory.

iv. Terminal name, Terminal type.

v. Machine name.

vi. No. of users who are currently logged in; List of users who are currently logged in.

```
echo 1. Calendar of the current month and year.
echo 2. Display Good Morning/Good Afternoon/Good Evening according to the current login time.
echo 3. User name, Users home directory.
echo 4. Terminal name, Terminal type.
echo 5 Machine name.
echo 6. No. of users who are currently logged in List of users who are currently logged in.
echo enter your choice
read ch
case $ch in
1) echo Calendar of current month is
cal ;;
2) d=`date +"%H"`
```

```
if [ $d -lt 12 ]
then
echo Good Morning
elif [ $d -gt 12 -a $d -lt 16 ]
then
echo Good Afternoon
else
echo Good Evening
fi
;;
3) echo Username is $USER
echo Users Home directory is $HOME ;;
4) echo Terminal details
tty;;
5) echo Machine name is
uname -m ;;
6) echo The number of users logged in are
who | wc -l
*) echo Invalid choice
esac
```

20 Write a shell script that displays the following menu and acts accordingly

1. Concatenates two strings

2. Renames a file

3. Deletes a file.

4. Copy the file to specific location

```
echo 1. Concatenates two strings
echo 2. Renames a file
echo 3. Deletes a file.
echo 4. Copy the file to specific location
echo enter your choice
read ch
case $ch in
```

```
1) echo enter first string
read str1
echo enter second string
read str2
echo The concated strings are $str1$str2 ;;
2) echo enter the old filename
read ofn
echo enter the new filename
read nfn
mv $ofn $nfn
echo file renamed ;;
3) echo enter the filename
read fn
rm $fn
echo file deleted ;;
4) echo enter the filename
read fn
cp $fn \usr\home\dir\$fn #you can change the specific path
echo file copied ;;
*) echo invalid choice ;;
esac
```