OSTL 2

1. echo $HOME, $PATH
   echo $MAIL
   echo $USER, $SHELL, $TERM

```
$ echo $HOME, $PATH
/home/student, /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
$ echo $MAIL

$ echo $USER, $SHELL, $TERM
student, /bin/sh, xterm-256color
$
```

2.

```
$ firstname=Rakesh
$ lastname=Sharma
$ echo $firstname $lastname
Rakesh Sharma
$ export lastname
$ sh echo $firstname $lastname
sh: 0: Can't open echo
$ sh
$ echo $firstname $lastname
Sharma
$
$ echo $firstname $lastname
Rakesh Sharma
$
```

3.

```
$ cat > script.sh
echo the name of this script is $0
echo the first argument is $1
echo a list of all arguments is $*
echo This script places the date into a temporary file
echo called $1.$$
date > $1.$$
ls $1.$$
rm $1.$$
$ chmod +x script.sh
$ ./script.sh rahul Sachin Kumble
the name of this script is ./script.sh
the first argument is rahul
a list of all arguments is rahul Sachin Kumble
This script places the date into a temporary file
called rahul.3660
rahul.3660
```

4.

```
$ (sleep 10; echo done)&
$ ps
  PID TTY          TIME CMD
 2919 pts/0    00:00:00 sh
 3749 pts/0    00:00:00 sh
 3750 pts/0    00:00:00 sleep
 3751 pts/0    00:00:00 ps
$ done
```

5.

```
$ (sleep 20; echo done)&
$ ps
  PID TTY          TIME CMD
 2919 pts/0    00:00:00 sh
 3820 pts/0    00:00:00 sh
 3821 pts/0    00:00:00 sleep
 3822 pts/0    00:00:00 ps
$ kill 3821
$ Terminated
done
```

6.

```
$ (sleep 30; echo done 1)&
$ (sleep 30; echo done 2) &
$ echo done 3; wait ; echo done 4
done 3
done 1
done 2
done 4
[2] + Done                       (sleep 30; echo done 2)
[1] + Done                       (sleep 30; echo done 1)
$
```

Exercise 2

1. List all the files under the given input directory whose extension has only one character.

```
$ cat > file1.c
ghjhjuj
$ cat > file2.c
trbhjjk
^C
$ cat > file4.h
hjhjjn
$ ls
200905090_Atul  Downloads         file2.c      hello.java    lab1_1.java   lab1q1  Pictures   snap
Desktop         examples.desktop  file4.h      lab1          lab1_3.class  lab1q2  Public     Templates
Documents       file1.c           hello.class  lab1_1.class  lab1_3.java   Music   script.sh  Videos
$ ls *.?
file1.c  file2.c  file4.h
```

2. Write a shell script that accepts 2 command line parameters. First parameter indicates the directory and the second parameter indicates a regular expression. The script should display all the files and directories in the directory specified in the first argument matching the format specified in the second argument.

```
$ mkdir files
$ cd files
$ cat > f1.txt
cgfbhj
ghvhjbhj
$ cat > f2.txt
fhyjk
vhjhjjk
$ cat > hello.c
jjk
hjhjbjk
$ ls
f1.txt  f2.txt  hello.c
$ cd ..
$ cat > file.sh
cd $1
ls *.$2
$ chmod +x file.sh
$ ./file.sh files txt
f1.txt  f2.txt
```

3. Count the number of users logged on to the system. Display the output as Number of users logged into the system.

```
$ echo The number of users logged into the system is `who | wc -l`
The number of users logged into the system is 1
```

4. Count only the number of files in the current directory.

```
$ echo the number of files in the current directory is `ls -l | grep "^-" | wc -l`
the number of files in the current directory is 10
$ ls
E1.png  e21.png  e22.png  e23.png  e2.png  e3.png  e4.png  e5.png  e6.png  files  file.sh
$
```

5. Write a shell script that takes 2 sorted numeric files as input and produces a single sorted numeric file without any duplicate contents.

```
$ cat > nf1
1
2
3
$ cat > nf2
2
3
4
$ cat > sorting.sh
sort -un $1 $2 > output.txt
$ chmod +x sorting.sh
$ ./sorting.sh nf1 nf2
$ cat output.txt
1
2
3
4
$
```

6. Write a shell script that accepts 2 command line arguments. First argument indicates format of file and second argument indicates the destination directory. The script should copy all the files as specified by the first argument to the location indicated by the second argument. Also, try the script where the destination directory name has a space in it.

```
$ ls
E1.png  e21.png  e22.png  e23.png  e24.png  e25.png  e2.png  e3.png  e4.png  e5.png  e6.png  files  file.sh  nf1  nf2  output.txt  sorting.sh
$ cat > copy.sh
cp *.$1 $2
$ chmod +x copy.sh
$ ./copy.sh png files
$ cd files
$ ls
dir  E1.png  e21.png  e22.png  e23.png  e24.png  e25.png  e2.png  e3.png  e4.png  e5.png  e6.png  f1.txt  f2.txt  hello.c
```