Fr. Conceicao Rodrigues College of Engineering Department of Computer Engineering			
Student's Roll No	**9913**	**Students Name**	**Mark Lopes**
Date of Performance	**01.02.2024**	**SE Computer – Div**	**A**

Aim: To study basics of Shell Scripting

Lab Outcome:
CSL403.1: Demonstrate basic Operating system Commands, Shell scripts, System Calls and API wrt Linux.

Problem Statements:

1. WAP that accepts user name and reports if user logged in.

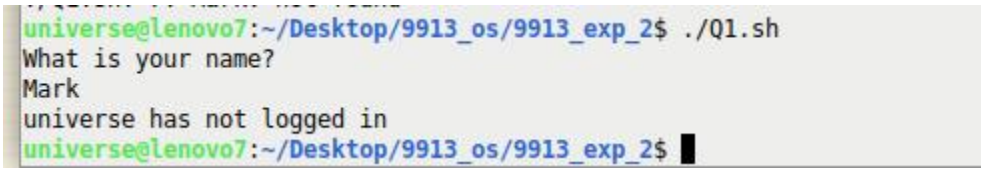
   CODE:

```
#!/bin/sh

printf "What is your name?\n"
read username

[[ $username == $USER ]] && echo "$username has logged in" && exit 0
echo "$username has not logged in"
exit 1
```

OUTPUT:



2. WAP that takes a filename as input and checks if it is executable, if not make it executable.

   CODE:

```
#!/bin/bash
printf "Please enter the file name: "
read filename
[[ -x $filename ]] && echo "The file is executable" && exit
0
echo "The file is not executable"
```

```
exit 1
```

OUTPUT:

```
universe@lenovo7:~/Desktop/9913_os/9913_exp_2$ ls
Q1.sh  Q2.sh  Q3.sh  Q4.sh  Q6.sh
universe@lenovo7:~/Desktop/9913_os/9913_exp_2$ ./Q2.sh
Enter the filename?
Q1.sh
File is executable
universe@lenovo7:~/Desktop/9913_os/9913_exp_2$
```

3. WAP to take string as command line argument and reverse it.

CODE:

```
#!/bin/bash
printf "Enter a string "
read str
printf "Reverse is : " && echo "$str" | rev
```

OUTPUT:

```
universe@lenovo6:~$ cd Desktop
universe@lenovo6:~/Desktop$ bash lab2_OS.sh mdlkmfef
Reversed string: fefmkldm
universe@lenovo6:~/Desktop$
```

4. Write a shell script to find the sum, the average and the product of the four integers entered

CODE:

```
#!/bin/bash
sumOfNumbers=0
product=1
for i in {1..4}; do
 printf "Enter a number\n"
 read num
 sumOfNumbers=$(( sumOfNumbers +  num ))
 product=$(( product * num ))
done
echo -e "The sum is ${sumOfNumbers}.\nThe product is ${product}.\nThe average
of the numbers is $(( sumOfNumbers / 4 ))"
```

OUTPUT:

```
universe@lenovo7:~/Desktop/9913_os/9913_exp_2$ ./Q4.sh
Enter number
4
Enter number
1
Enter number
2
Enter number
3
The sum is 10
The product is 24
The average is 2
```

5. Write a shell script to find out the unique words in a file and also count the occurrence of each of these words. We can say that the file under consideration contains many lines, and each line has multiple words.

```
unique=$(bat firstFile.txt | tr ' ' '\n' | sort | uniq)
uniqueCount=$(bat firstFile.txt | tr ' ' '\n' | sort | uniq | wc -l)
echo -e "The count of unique words is: ${uniqueCount}\n"
echo -e "Word\t\t Count"
for i in $unique; do
  printf "$i\t\t $(bat firstFile.txt | tr ' ' '\n' | rg "$i" | wc -l)\n"
done
```

OUTPUT:

```
The count of unique words is : 9

Word            Count
?               0
am              1
are             1
Fine            1
Hello           1
How             1
I               1
today           1
you             1
```

14. WAP which displays the following menu and executes the option selected by user:

    1.ls 2. pwd 3. ls –l 4. ps -fe

    CODE:

```
#!/bin/bash
echo -e "Menu:\n1.ls\n2.pwd\n3.ls -l\n4.ps -fe"
echo -e "Please select one from the menu(1 | 2 | 3| 4):"
read ans
case "$ans" in
 1)
   echo "The output of ls is:"
   ls;;
 2)
   echo "The output of pwd is:"
   pwd;;
 3)
   echo "The output of ls -l is:"
```

```
   ls -l;;
 4)
   echo "The output of ps -fe is:"
   ps -fe;;
 *)
   echo "Please select a valid option"
   ;;
esac
```

OUTPUT

```
universe@lenovo7:~/Desktop/9913_os/9913_exp_2$ ./Q6.sh
-e Menu:-
1.ls
2.pwd
3.ls -l
4.ps -fe
Select any one(1,2,3,4)
1
ls:
Q1.sh  Q2.sh  Q3.sh  Q4.sh  Q6.sh
universe@lenovo7:~/Desktop/9913_os/9913_exp_2$ ./Q6.sh
-e Menu:-
1.ls
2.pwd
3.ls -l
4.ps -fe
Select any one(1,2,3,4)
2
pwd:
/home/universe/Desktop/9913_os/9913_exp_2
```

```
universe@lenovo7:~/Desktop/9913_os/9913_exp_2$ ./Q6.sh
-e Menu:-
1.ls
2.pwd
3.ls -l
4.ps -fe
Select any one(1,2,3,4)
3
ls -l:
total 20
-rwxrwxr-x 1 universe universe  78 Feb  1 14:42 Q1.sh
-rw-rw-r-- 1 universe universe 136 Feb  1 15:10 Q2.sh
-rwxrwxr-x 1 universe universe 145 Feb  1 15:20 Q3.sh
-rwxrwxr-x 1 universe universe 203 Feb  1 15:33 Q4.sh
-rwxrwxr-x 1 universe universe 260 Feb  1 15:45 Q6.sh
```

```
universe@lenovo7:~/Desktop/9913_os/9913_exp_2$ ./Q6.sh
-e Menu:-
1.ls
2.pwd
3.ls -l
4.ps -fe
Select any one(1,2,3,4)
4
ps -fe:
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 13:46 ?        00:00:01 /sbin/init splash
root           2       0  0 13:46 ?        00:00:00 [kthreadd]
root           3       2  0 13:46 ?        00:00:00 [rcu_gp]
root           4       2  0 13:46 ?        00:00:00 [rcu_par_gp]
root           5       2  0 13:46 ?        00:00:00 [netns]
root           7       2  0 13:46 ?        00:00:00 [kworker/0:0H-events_highpri
root          10       2  0 13:46 ?        00:00:00 [mm_percpu_wq]
root          11       2  0 13:46 ?        00:00:00 [rcu_tasks_rude_]
root          12       2  0 13:46 ?        00:00:00 [rcu_tasks_trace]
root          13       2  0 13:46 ?        00:00:00 [ksoftirqd/0]
root          14       2  0 13:46 ?        00:00:02 [rcu_sched]
root          15       2  0 13:46 ?        00:00:00 [migration/0]
root          16       2  0 13:46 ?        00:00:00 [idle_inject/0]
root          17       2  0 13:46 ?        00:00:00 [cpuhp/0]
root          18       2  0 13:46 ?        00:00:00 [cpuhp/1]
root          19       2  0 13:46 ?        00:00:00 [idle_inject/1]
root          20       2  0 13:46 ?        00:00:00 [migration/1]
root          21       2  0 13:46 ?        00:00:00 [ksoftirqd/1]
root          23       2  0 13:46 ?        00:00:00 [kworker/1:0H-events_highpri
root          24       2  0 13:46 ?        00:00:00 [cpuhp/2]
root          25       2  0 13:46 ?        00:00:00 [idle_inject/2]
root          26       2  0 13:46 ?        00:00:00 [migration/2]
root          27       2  0 13:46 ?        00:00:00 [ksoftirqd/2]
root          29       2  0 13:46 ?        00:00:00 [kworker/2:0H-kblockd]
root          30       2  0 13:46 ?        00:00:00 [cpuhp/3]
root          31       2  0 13:46 ?        00:00:00 [idle_inject/3]
root          32       2  0 13:46 ?        00:00:00 [migration/3]
root          33       2  0 13:46 ?        00:00:00 [ksoftirqd/3]
root          35       2  0 13:46 ?        00:00:00 [kworker/3:0H-events_highpri
root          36       2  0 13:46 ?        00:00:00 [cpuhp/4]
root          37       2  0 13:46 ?        00:00:00 [idle_inject/4]
root          38       2  0 13:46 ?        00:00:00 [migration/4]
root          39       2  0 13:46 ?        00:00:00 [ksoftirqd/4]
root          41       2  0 13:46 ?        00:00:00 [kworker/4:0H-kblockd]
root          42       2  0 13:46 ?        00:00:00 [cpuhp/5]
```

15. WAP that prompts user for a starting value & counts down from there.

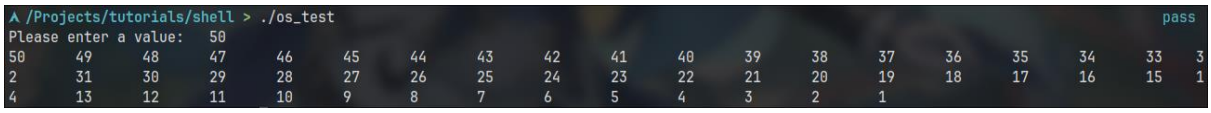
CODE

```
#!/bin/bash

# Prompt the user to enter a starting value
echo "Enter a starting value:"
read start_value

# Count down from the starting value
echo -n "$start_value"
while [ $start_value -gt 0 ]; do
    ((start_value--))
    echo -ne "\t$start_value"
done
echo
```

OUTPUT:



16. Create a data file called employee in the format given below:

    EmpCode Character, EmpName Character , Grade Character , Years of experience Numeric , Basic Pay Numeric

    - Sort the file on EmpName.
    - Sort the file on
      Decreasing order of basic pay
      Increasing order of years of experience.
      Display the number of employees whose details are included in the file.
      Display all records with ‘smith’ a part of employee name.
      Display all records with EmpName starting with ‘B’.

CODE:

```
#!/bin/bash
if ! [[ -f employee.txt ]]; then
 echo "EmpCode,EmpName,Grade,Years of experience,Basic Pay"
>employee.txt
 echo "E001,John Doe,A,5,50000" >>employee.txt
 echo "E002,Jane Smith,B,3,45000" >>employee.txt
 echo "E003,Bob Johnson,C,8,60000" >>employee.txt
 echo "E004,Alice Williams,A,6,52000" >>employee.txt
 echo "E005,Charlie Brown,D,2,40000" >>employee.txt
 echo "E006,Emily smith,B,4,48000" >>employee.txt
 echo "E007,David Miller,C,7,55000" >>employee.txt
 echo "E008,Emma White,A,9,62000" >>employee.txt
 echo "E009,Frank Thompson,D,1,38000" >>employee.txt
 echo "E010,Grace Turner,B,5,51000" >>employee.txt
 echo "E011,Henry Harris,C,3,47000" >>employee.txt
 echo "E012,Isabel Clark,A,8,59000" >>employee.txt
 echo "E013,Jack Turner,D,2,42000" >>employee.txt
 echo "E014,Karen Anderson,B,4,49000" >>employee.txt
 echo "E015,Liam Smith,C,6,54000" >>employee.txt
 echo "E016,Mia Robinson,A,7,56000" >>employee.txt
 echo "E017,Noah Walker,D,5,53000" >>employee.txt
 echo "E018,Olivia Harris,B,2,43000" >>employee.txt
 echo "E019,Peter Turner,C,4,50000" >>employee.txt
 echo "E020,Quinn Adams,A,1,40000" >>employee.txt
fi
```

```
if ! [[ -f emp_name.txt ]]; then
 # sort the keys seperated by comma from 2nd column to 2nd column
 sort -t ',' -k 2,2 -o emp_name.txt employee.txt
fi

if ! [[ -f emp_salary.txt ]]; then
 # sort the 5 row by comparing the numeric value and print in
reverse
 sort -t ',' -k 5,5 -nr -o emp_salary.txt employee.txt
fi

if ! [[ -f emp_year.txt ]]; then
 # sort the 5 row by comparing the numeric value and print in
reverse
 sort -t ',' -k 4,4 -n -o emp_year.txt employee.txt
fi

# or we can do $(($wc -l employee.txt | awk '$1-=1;{print $1}) -
1 ))'
echo "The number of employees are: $(($(bat employee.txt | wc -l)
- 1))"
# just grep the case insensitive smith word
echo -e "\nThe employees who have 'smith' in thier name are\n
$(bat employee.txt | rg -i smith)"
# basically some regex magic that cannot be explained in simple
words.
echo -e "\nThe employees whose name starts with 'B' are\n $(bat
employee.txt | rg ".*[E0-9],B.*\ .*,[A-E],.*[0-9],,*[0-9]")"
```

OUTPUT:

```
/Projects/tutorials/shell > ./os_test
The number of employees are: 20

The employees who have 'smith' in thier name are
 E002,Jane Smith,B,3,45000
E006,Emily smith,B,4,48000
E015,Liam Smith,C,6,54000

The employees whose name starts with 'B' are
 E003,Bob Johnson,C,8,60000
```

File: emp_name.txt

```
1  E004,Alice Williams,A,6,52000
2  E003,Bob Johnson,C,8,60000
3  E005,Charlie Brown,D,2,40000
4  E007,David Miller,C,7,55000
5  E006,Emily smith,B,4,48000
6  E008,Emma White,A,9,62000
7  EmpCode,EmpName,Grade,Years of experience,Basic Pay
8  E009,Frank Thompson,D,1,38000
9  E010,Grace Turner,B,5,51000
10 E011,Henry Harris,C,3,47000
11 E012,Isabel Clark,A,8,59000
12 E013,Jack Turner,D,2,42000
13 E002,Jane Smith,B,3,45000
14 E001,John Doe,A,5,50000
```

File: emp_salary.txt

```
E008,Emma White,A,9,62000
E003,Bob Johnson,C,8,60000
E012,Isabel Clark,A,8,59000
E016,Mia Robinson,A,7,56000
E007,David Miller,C,7,55000
E015,Liam Smith,C,6,54000
E017,Noah Walker,D,5,53000
E004,Alice Williams,A,6,52000
E010,Grace Turner,B,5,51000
E019,Peter Turner,C,4,50000
E001,John Doe,A,5,50000
E014,Karen Anderson,B,4,49000
```

File: emp_year.txt

```
EmpCode,EmpName,Grade,Years of experience,Basic Pay
E009,Frank Thompson,D,1,38000
E020,Quinn Adams,A,1,40000
E005,Charlie Brown,D,2,40000
E013,Jack Turner,D,2,42000
E018,Olivia Harris,B,2,43000
E002,Jane Smith,B,3,45000
E011,Henry Harris,C,3,47000
E006,Emily smith,B,4,48000
E014,Karen Anderson,B,4,49000
E019,Peter Turner,C,4,50000
E001,John Doe,A,5,50000
E010,Grace Turner,B,5,51000
E017,Noah Walker,D,5,53000
```

References:

https://www.tutorialspoint.com/unix/shell_scripting.htm

https://www.tutorialspoint.com/unix/unix-what-is-shell.htm

https://www.tutorialspoint.com/unix/unix-using-variables.htm

https://www.tutorialspoint.com/unix/unix-special-variables.htm

https://www.tutorialspoint.com/unix/unix-using-arrays.htm

https://www.tutorialspoint.com/unix/unix-basic-operators.htm

https://www.tutorialspoint.com/unix/unix-decision-making.htm

https://www.tutorialspoint.com/unix/unix-shell-loops.htm

On time Submission(2)	Knowledge of Topic(4)	Implementation and Demonstraion(4)	Total (10)
Signature of Faculty		**Date of Submission**	