National University of Modern Languages

Assignment#02

Roll # 2340

Class: BSCS 5B Morning

Subject: Operating System(Lab)

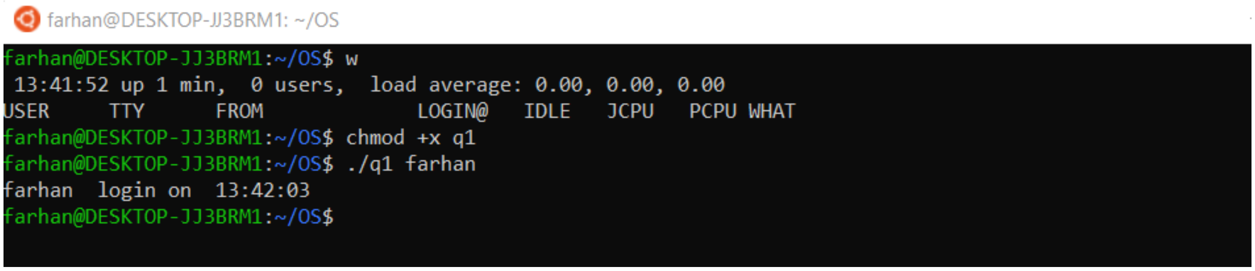
Submitted to: Mrs. Humaira Batool

Submitted by: Farhan Haider

Q1: Write a shell script that takes a login name as a command-line argument and reports when that person logs in?

```
#!/bin/bash

name=$(whoami)

if [ $name = $1 ]

then

time=`w -s | cut -d ' ' -f2`

echo $name " login on  "$time

else

echo "not login"

fi

exit 0;
```

OUTPUT:

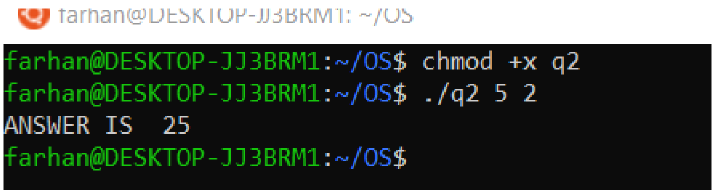


```
farhan@DESKTOP-JJ3BRM1:~/OS$ w
 13:41:52 up 1 min,  0 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
farhan@DESKTOP-JJ3BRM1:~/OS$ chmod +x q1
farhan@DESKTOP-JJ3BRM1:~/OS$ ./q1 farhan
farhan  login on  13:42:03
farhan@DESKTOP-JJ3BRM1:~/OS$
```

Q2: Write a shell script that accepts two integers as its arguments and computes the value of the first number raised to the power of the second number.

```
#!/bin/bash

Expansion(){

x=$(($1**$2))

echo "ANSWER IS " $x

}

Expansion $1 $2

exit 0;
```

OUTPUT:

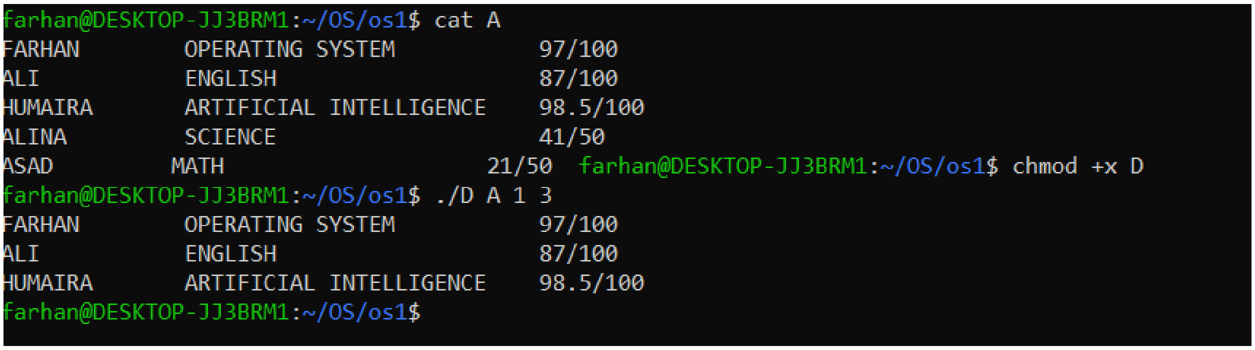


```
farhan@DESKTOP-JJ3BRM1: ~/OS
farhan@DESKTOP-JJ3BRM1:~/OS$ chmod +x q2
farhan@DESKTOP-JJ3BRM1:~/OS$ ./q2 5 2
ANSWER IS  25
farhan@DESKTOP-JJ3BRM1:~/OS$
```

Q3: Write a shell script that accepts a filename, starting and ending line numbers as arguments, and displays all the lines between the given line numbers.

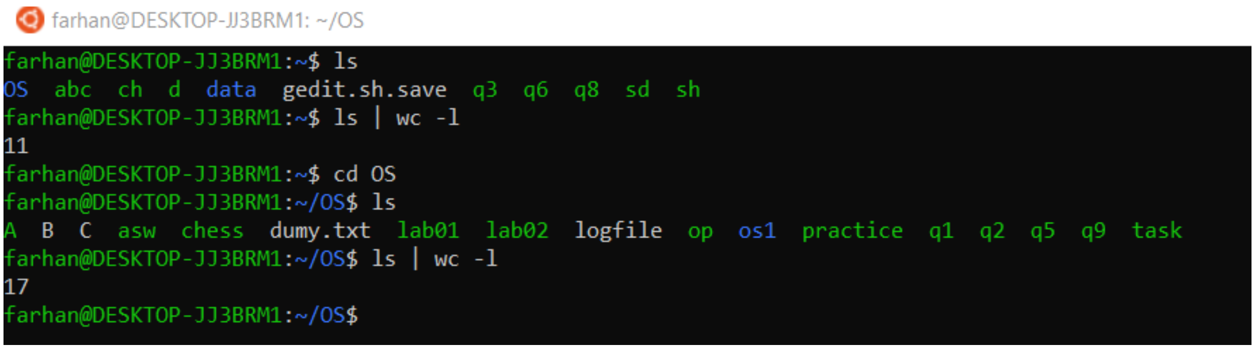
```
#!/bin/bash

sed -n $2,$3\p $1 | cat > newline

cat newline

exit 0;
```

OUTPUT:



```
farhan@DESKTOP-JJ3BRM1:~/OS/os1$ cat A
FARHAN         OPERATING SYSTEM          97/100
ALI            ENGLISH                   87/100
HUMAIRA        ARTIFICIAL INTELLIGENCE   98.5/100
ALINA          SCIENCE                   41/50
ASAD          MATH                   21/50  farhan@DESKTOP-JJ3BRM1:~/OS/os1$ chmod +x D
farhan@DESKTOP-JJ3BRM1:~/OS/os1$ ./D A 1 3
FARHAN         OPERATING SYSTEM          97/100
ALI            ENGLISH                   87/100
HUMAIRA        ARTIFICIAL INTELLIGENCE   98.5/100
farhan@DESKTOP-JJ3BRM1:~/OS/os1$
```

Q4: Write a shell script To Count number of files in a Directory.



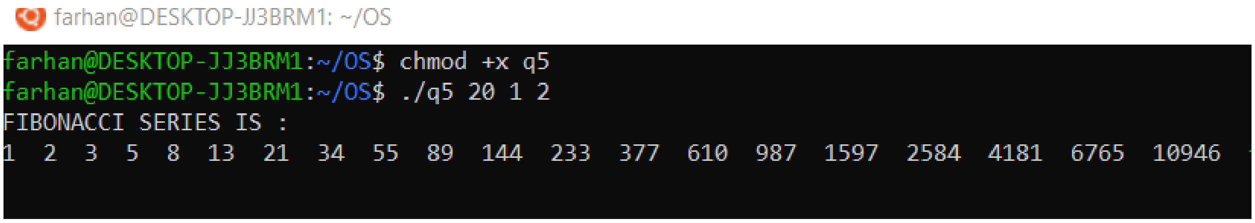
```
farhan@DESKTOP-JJ3BRM1:~$ ls
OS  abc  ch  d  data  gedit.sh.save  q3  q6  q8  sd  sh
farhan@DESKTOP-JJ3BRM1:~$ ls | wc -l
11
farhan@DESKTOP-JJ3BRM1:~$ cd OS
farhan@DESKTOP-JJ3BRM1:~/OS$ ls
A  B  C  asw  chess  dumy.txt  lab01  lab02  logfile  op  os1  practice  q1  q2  q5  q9  task
farhan@DESKTOP-JJ3BRM1:~/OS$ ls | wc -l
17
farhan@DESKTOP-JJ3BRM1:~/OS$
```

Q5: Write a program to generate the Fibonacci series.

```
#!/bin/bash

n=$1

a=$2

b=$3

echo "FIBONACCI SERIES IS : "

for (( i=0; i<n; i++ ))

do
```

```
echo -n "$a" " "

fn=`expr $a + $b`

a=$b

b=$fn

done

exit 0;
```

OUTPUT:



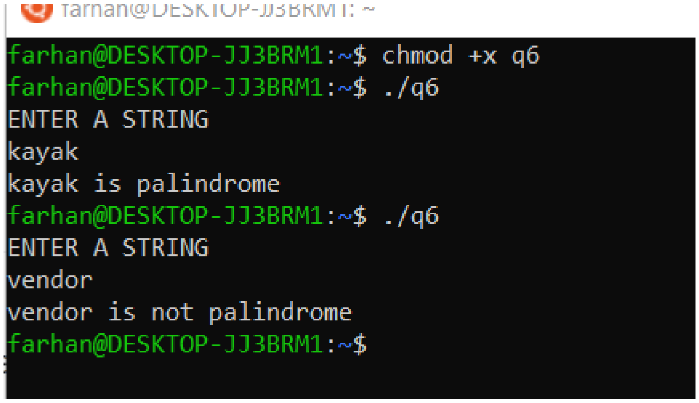
```
farhan@DESKTOP-JJ3BRM1:~/OS$ chmod +x q5
farhan@DESKTOP-JJ3BRM1:~/OS$ ./q5 20 1 2
FIBONACCI SERIES IS :
1  2  3  5  8  13  21  34  55  89  144  233  377  610  987  1597  2584  4181  6765  10946
```

Q6: Write a program to check whether given string is palindrome or not.

```
#!/bin/bash

echo "ENTER A STRING"

read str

revserse=""

length=${#str}

for (( i=$length-1; i>=0; i-- ))

do

reverse=$reverse${str:$i:1}

done
```

```
if [ $str == $reverse ]

then

echo "$str is palindrome"

else

echo "$str is not palindrome"

fi

exit 0;
```

OUTPUT:



```
farhan@DESKTOP-JJ3BRM1: ~
farhan@DESKTOP-JJ3BRM1:~$ chmod +x q6
farhan@DESKTOP-JJ3BRM1:~$ ./q6
ENTER A STRING
kayak
kayak is palindrome
farhan@DESKTOP-JJ3BRM1:~$ ./q6
ENTER A STRING
vendor
vendor is not palindrome
farhan@DESKTOP-JJ3BRM1:~$
```

Q7: Write a shell program for the following Scenario:

Enter any Job Title

1. Tutor

2. Lecturer

3. Associate Professor

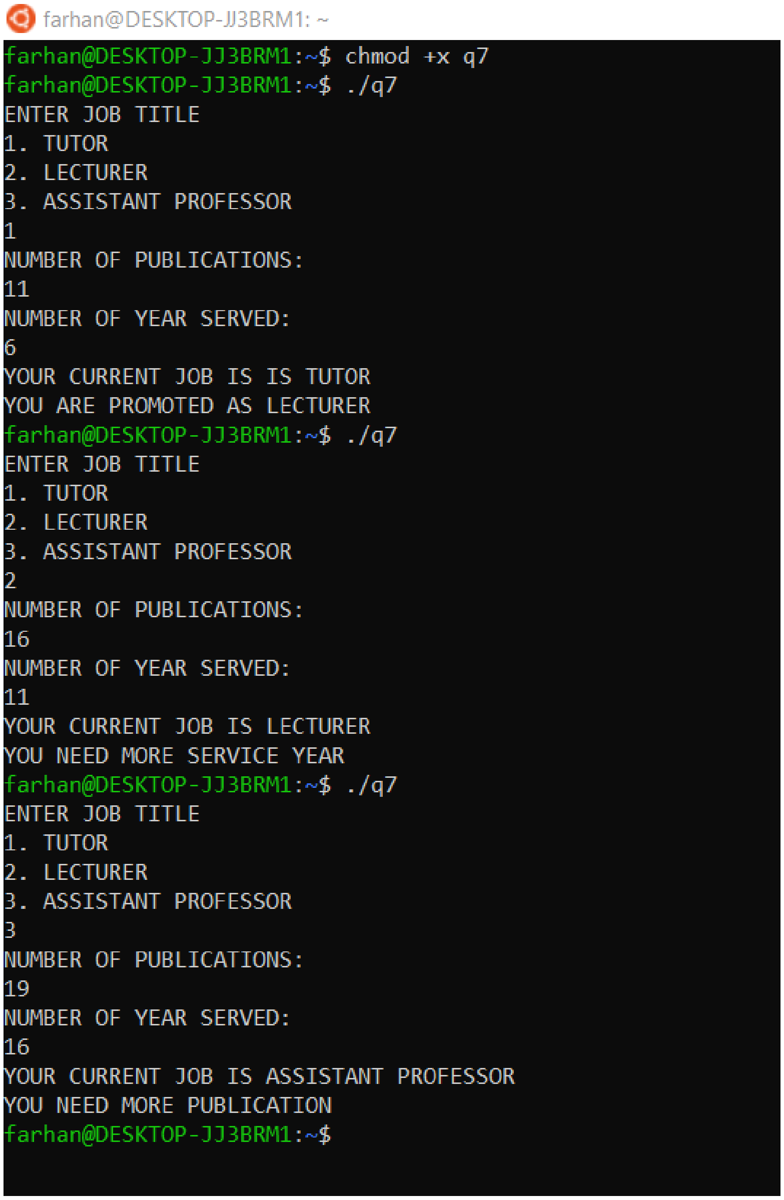
If years served greater than 5 and publication greater than 10, then promote the person as a lecturer. And if years served greater than 12 and publications greater than 15, then promote the person as Associate professor. And If years served greater than 15 and publications greater than 20, then promote the person as a professor else more service and publications are required.

```
#!/bin/bash
echo "ENTER JOB TITLE"
echo "1. TUTOR"
echo "2. LECTURER"
echo "3. ASSISTANT PROFESSOR"
read X
echo "NUMBER OF PUBLICATIONS:"
read p
echo "NUMBER OF YEAR SERVED:"
read y
case $X in
1)
echo "YOUR CURRENT JOB IS IS TUTOR"
if [ $y -gt 5 ] && [ $p -gt 10 ]
then
echo "YOU ARE PROMOTED AS LECTURER"
elif [ $y -le 5 ]
then
echo "YOU NEED MORE SERVICE YEAR"
elif [ $p -le 10 ]
then
echo "YOU NEED MORE PUBLICATION"
```

```
fi
;;
2)
echo "YOUR CURRENT JOB IS LECTURER"
if [ $y -gt 12 ] && [ $p -gt 15 ]
then
echo "YOU ARE PROMOTED AS ASSISTANT PROFESSOR"
elif [ $y -le 12 ]
then
echo "YOU NEED MORE SERVICE YEAR"
elif [ $p -le 15 ]
then
echo "YOU NEED MORE PUBLICATION"
fi
;;
3)
echo "YOUR CURRENT JOB IS ASSISTANT PROFESSOR"
if [ $y -gt 15 ] && [ $p -gt 20 ]
then
echo "YOU ARE PROMOTED AS PROFESSOR"
elif [ $y -le 15 ]
then
```

```
echo "YOU NEED MORE SERVICE YEAR"

elif [ $p -le 20 ]

then

echo "YOU NEED MORE PUBLICATION "

fi

;;

*)

echo "WRONG JOB TITLE"

;;

esac

exit 0;
```

OUTPUT:



```
farhan@DESKTOP-JJ3BRM1:~$ chmod +x q7
farhan@DESKTOP-JJ3BRM1:~$ ./q7
ENTER JOB TITLE
1. TUTOR
2. LECTURER
3. ASSISTANT PROFESSOR
1
NUMBER OF PUBLICATIONS:
11
NUMBER OF YEAR SERVED:
6
YOUR CURRENT JOB IS IS TUTOR
YOU ARE PROMOTED AS LECTURER
farhan@DESKTOP-JJ3BRM1:~$ ./q7
ENTER JOB TITLE
1. TUTOR
2. LECTURER
3. ASSISTANT PROFESSOR
2
NUMBER OF PUBLICATIONS:
16
NUMBER OF YEAR SERVED:
11
YOUR CURRENT JOB IS LECTURER
YOU NEED MORE SERVICE YEAR
farhan@DESKTOP-JJ3BRM1:~$ ./q7
ENTER JOB TITLE
1. TUTOR
2. LECTURER
3. ASSISTANT PROFESSOR
3
NUMBER OF PUBLICATIONS:
19
NUMBER OF YEAR SERVED:
16
YOUR CURRENT JOB IS ASSISTANT PROFESSOR
YOU NEED MORE PUBLICATION
farhan@DESKTOP-JJ3BRM1:~$
```

Q8: Write a shell script to display the following pattern.

1 0

1 5

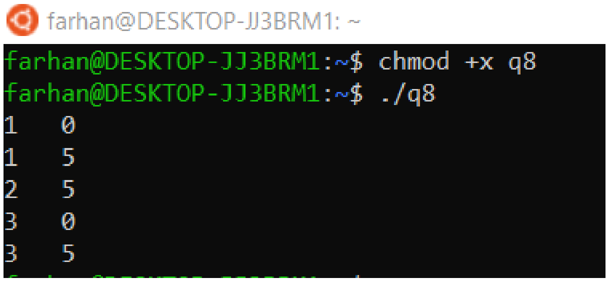
2 5

3 0

3 5

```
#!/bin/bash

for((i=1; i<=3; i++))

do

for((j=0; j<10; j=j+5))

do

if [ $i -eq 2  ] && [ $j -eq 0 ]

then

echo -n

else

echo $i " " $j

fi

done

done

exit 0;
```

OUTPUT:

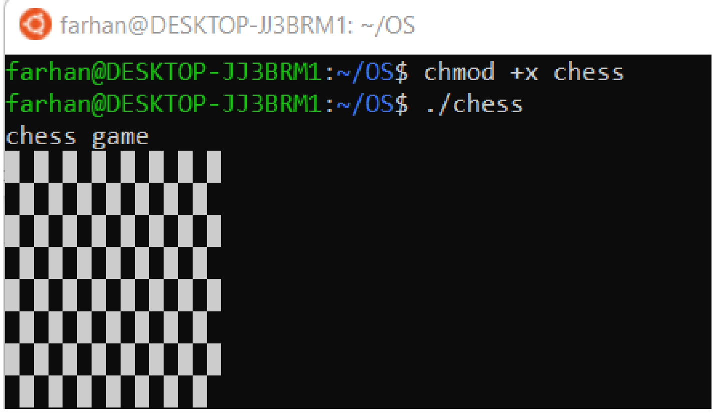


```
farhan@DESKTOP-JJ3BRM1:~$ chmod +x q8
farhan@DESKTOP-JJ3BRM1:~$ ./q8
1   0
1   5
2   5
3   0
3   5
```

Q9: Write a shell script to Display CHESSBOARD patterns.

```
#!/bin/bash
echo "chess game"
for (( i = 1; i <= 8; i++ ))
do
   for (( j = 1 ; j <= 15; j++ ))
   do
        total=`expr $i + $j`
        tmp=`expr $total % 2`
        if [ $tmp -eq 0 ];
        then
            echo -e -n "\033[47m "
        else
            echo -e -n "\033[40m "
        fi
  done
 echo -e -n "\033[40m"
 echo ""
done
exit 0;
```

OUTPUT:



```
farhan@DESKTOP-JJ3BRM1: ~/OS
farhan@DESKTOP-JJ3BRM1:~/OS$ chmod +x chess
farhan@DESKTOP-JJ3BRM1:~/OS$ ./chess
chess game
```

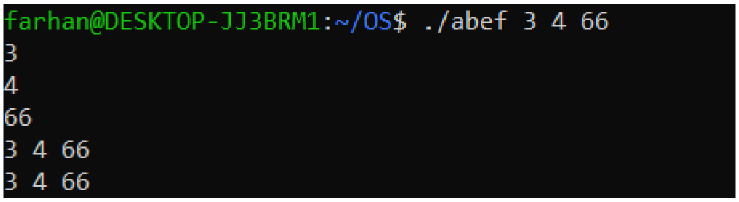
Q10: What is the difference between $* And $@?

$@ is nearly the same as $*, both meaning "all command-line arguments". They are often used to simply pass all arguments to another program $* expands to all parameters that were passed to that shell script.

FOR EXAMPLE:

```
#!/bin/bash

echo $1

echo $2

echo $3

echo $*

echo $@

exit 0;
```

OUTPUT:



```
farhan@DESKTOP-JJ3BRM1:~/OS$ ./abef 3 4 66
3
4
66
3 4 66
3 4 66
```