LABSHEET 4

1. Write a shell script to generate emails in the given format and write it into a file. Your script should accept sender and recipient email id's and subject as command line arguments.

From: abc@domain1.com

To: xx@domain.com

Cc: yy@domain.com

Subject: Subject 1

This email is generated by my shell script.

Thanks and regards

S4 CSE student

Amritapuri

```
echo "From : "
read From
echo "To :"
read To
echo "Cc: "
echo "Subject : "
read Subject

cat <<EOF >email
From : $From
To : $To
Cc : $Cc
Subject : $Subject
                This mail is generates by my shell script.
        Thanks and regards
        S4 CSE students
        Amritapuri
EOF
cat email
```

```
From :
abc@domain.com
To :
xyz@domain2.com
Cc:
yy@domain2.com
Subject :
Subject
From : abc@domain.com
To : xyz@domain2.com
Cc : yy@domain2.com
Subject : Subject
                This mail is generates by my shell script.
        Thanks and regards
        S4 CSE students
        Amritapuri
```

2. Modify Question 1 to allow user to enter text at the beginning of email content, by passing it as a command line argument.

```
echo "From : "
read From
echo "To :"
read To
echo "Cc: "
read Cc
echo "Subject : "
read Subject
echo "Content :"
read Content
cat <<EOF >email
From : $From
To : $To
Cc : $Cc
Subject : $Subject
                This mail is generates by my shell script.
        Thanks and regards
        S4 CSE students
        Amritapuri
EOF
```

```
From :
abc@gmail.com
To :
xyz@gmail.com
Cc :
ijk@gmail.com
Subject :
Greeting!!
Content :
Hello All..
From : abc@gmail.com
To : xyz@gmail.com
Cc : ijk@gmail.com
Subject : Greeting!!
                Hello All..
        Thanks and regaerds
        S4 CSE student
        Amritapuri
```

3. Write a shell script to print all the primes below a given number.

```
echo "Enter a number :"
read num

echo "Primes below $num"
for (( i==2;i<num;i++ ));do
        prime=1
        for (( j=2;j*j<=i;j++ ));do
                if (( i % j ==0 ));then
                        prime=0
                        break
                fi
        done
        if (( prime == 1 )); then
                echo -n "$i "
        fi
done
echo
```

```
ubuntu@ubuntu:~$ chmod 744 lab4.sh
ubuntu@ubuntu:~$ ./lab4.sh
Enter a number :
10
Primes below 10
 1 2 3 5 7
```

4. Write a shell script to print the first n Fibonacci numbers .

```
#4
echo "Enter number of Fibonacci terms :"
read n
a=0
b=1
echo "Fibonacci sequence: "

for (( i=0;i<n;i++ ));do
         echo -n "$a "
         temp=$((a+b))
         a=$b
         b=$temp
done
echo
```

```
Enter number of Fibonacci terms :
8
Fibonacci sequence:
0 1 1 2 3 5 8 13
```

5. Write a shell script to generate a multiplication table.

a. Interactive version: The program should accept an integer n given by the user and should print the multiplication table of that n.

```
read -p "Enter a number " num
for (( i=1;i<=10;i++ ));do
        echo "$num x $i =$((num * i ))"
done
```

```
Enter a number 5
5 x 1 =5
5 x 2 =10
5 x 3 =15
5 x 4 =20
5 x 5 =25
5 x 6 =30
5 x 7 =35
5 x 8 =40
5 x 9 =45
5 x 10 =50
```

b. Command line arguments version: The program should take the value of n from the arguments followed by the command.

```
if [ $# -ne 1 ];then
        echo "Usage : $0 <number>"
        exit 1
fi
num=$1
for (( i=1; i<=10;i++ ));do
        echo "$num x $i = $((num *i ))"
done
```

```
ubuntu@ubuntu:~$ ./lab_4.sh 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

c. Redirection version: The value of n must be taken from a file using input redirection.

```
Enter a number :
7
7 x 1= 7
7 x 2= 14
7 x 3= 21
7 x 4= 28
7 x 5= 35
7 x 6= 42
7 x 7= 49
7 x 8= 56
7 x 9= 63
7 x 10= 70
```

```
#5 c
echo "Enter a number : "
read num
echo $num > input.txt
read num < input.txt
for (( i=1 ; i<=10;i++ ));do
        echo "$num x $i= $((num *i))"
done
```

6. Using function write a shell script to find gcd of two numbers.

```
#6
read -p "Enter 2 numbers : " num1 num2
while [ $num2 -ne 0 ]; do
        temp=$num2
        num2=$((num1%num2))
        num1=$temp
done
echo "GCD is $num1"
```

```
Enter 2 numbers : 4 5
GCD is 1
```

7. Using Recursion find factorial of a number.

```
#7
read -p "Enter a number : " num
fact=1
i=$num
while [ $i -gt 1 ];do
        fact=$((fact*i))
        i=$((i-1))
done
echo "Factorial of $num : $fact "
```

```
Enter a number : 5
Factorial of 5 : 120
```

8. Write shell script to show various system configuration like:

a. Currently logged user and his long name

b. Current shell

c. Home directory

d. Operating system type

e. Current path setting

f. Current working directory

g. All available shells

```
#8
echo "System Configuration Details : "
echo "---------------------------------"
#a
echo "Logged-in User : $USER"
echo "Full Name : $(getent passwd "$USER" | cut -d ':' -f 5)"
#b
echo "Current Shell : $SHELL"
#c
echo "Home Directory : $HOME"
#d
echo "OS Type : $(uname -o)"
#e
echo "Current Path : $PATH"
#f
echo "Current Working Directory : $(pwd)"
#g
echo "Available Shells : "
cat /etc/shells
```

```
System Configuration Details :
---------------------------------
Logged-in User : ubuntu
Full Name : Live session user,,,
Current Shell : /bin/bash
Home Directory : /home/ubuntu
OS Type : GNU/Linux
Current Path : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr
/games:/usr/local/games:/snap/bin:/snap/bin
Current Working Directory : /home/ubuntu
Available Shells :
# /etc/shells: valid login shells
/bin/sh
/usr/bin/sh
```