IMPLEMENTATION

/*
Roll No : B21CSB69
Name : Sreelal V
Experiment No : 1.2
*/

1. → touch empty1 empty2 empty3 empty4 empty5

2. → cat > text
Sreelal
21
Malappuram

3. → cat text
Sreelal
21
Malappuram

4. → cp text newtext
5. → cat > maths
hello
world
hai
6. → paste maths text >textmat
7. → rm text
8. → chmod 666 newtext
9. → mv newtext oldtext
10. → mkdir mydir
11. → mv oldtext maths mydir
12. → mkdir mydir/newdir
13. → cp -r newdir mydir
14. → rm -i empty*
rm: remove regular empty file 'empty1'? y
rm: remove regular empty file 'empty2'? y
rm: remove regular empty file 'empty3'? y
rm: remove regular empty file 'empty4'? y
rm: remove regular empty file 'empty5'? y

IMPLEMENTATION

/*
Roll No : B21CSB69
Name : Sreelal V
Experiment No : 1.3
*/

1. → cat > fruits
Apple
banana
grapes
orange
mango
→ cp fruits newfruits

2. → sort -r fruits > reverse && cat reverse
orange
mango
grapes
banana
apple

3. → ls -l | wc -wl | tee count
5 38

4. → cat > poem
this is a beautiful poem
it is beautiful
I like my poem
poem name is poem
→ grep poem poem | wc -l | tee poemcount
3

5. → ls p* | tee two
poem
poemcount

6. → touch detail1 detail2
→ cat > detail1
Sreelal 69 Malappuram
→ cat > detail2
r4b tkmce
→ cat detail1 detail2 | tee detail
Sreelal 69 Malappuram
r4b tkmce

7. → cat > a.txt
aksdjklthaghalksthhtht a;djgakdhtht
hello this is a sample text
THIS IS SAMPLE TEXT IS CAPS

→ cat > b.txt
search a particular pattern ‘th’, ‘TH’, ‘Th’, ‘tH’
ths this theh ht tthh alksdrhaehtheth taed
thht akdfahthse thaethaetheht
→ cat a.txt b.txt > merged && sort merged | grep -i "th"
aksdjklthaghalksthhtht a;djgakdhtht
hello this is a sample text
search a particular pattern ‘th’, ‘TH’, ‘Th’, ‘tH’
thht akdfahthse thaethaetheht
THIS IS SAMPLE TEXT IS CAPS
ths this theh ht tthh alksdrhaehtheth taed

8. → ls -l | grep "Mar"
-rw-rw-r-- 1 hay hay 94 Mar 26 13:47 a.txt
-rw-rw-r-- 1 hay hay 140 Mar 26 13:48 b.txt
-rw-rw-r-- 1 hay hay 16 Mar 26 13:31 count
-rw-rw-r-- 1 hay hay 33 Mar 26 13:41 detail
-rw-rw-r-- 1 hay hay 23 Mar 26 13:41 detail1
-rw-rw-r-- 1 hay hay 10 Mar 26 13:41 detail2
-rw-rw-r-- 1 hay hay 33 Mar 26 13:21 fruits
-rw-rw-r-- 1 hay hay 234 Mar 26 13:56 merged
-rw-rw-r-- 1 hay hay 33 Mar 26 13:21 newfruits
-rw-rw-r-- 1 hay hay 74 Mar 26 13:35 poem
-rw-rw-r-- 1 hay hay 2 Mar 26 13:36 poemcount
-rw-rw-r-- 1 hay hay 33 Mar 26 13:28 reverse
-rw-rw-r-- 1 hay hay 15 Mar 26 13:38 two

9. → sleep 200& sleep 250& sleep 300& sleep 350&
[1] 5366
[2] 5367
[3] 5368
[4] 5369

10. → jobs
[1] Running sleep 200 &
[2] Running sleep 250 &
[3]- Running sleep 300 &
[4]+ Running sleep 350 &

→ fg 3
sleep 300
^Z
[3]+ Stopped sleep 300

→ bg
[3]+ sleep 300 &

IMPLEMENTATION

```
/*
Roll No         : B21CSB69
Name            : Sreelal V
Experiment No : 1.4
*/
```

a)

```
#!/bin/bash
a=20
b=10

echo "Sum of  $a + $b = $((a+b))"
echo "Sub of  $a - $b = $((a+b))"
echo "Mul of  $a * $b = $((a+b))"
echo "Div of  $a / $b = $((a+b))"
```

output:

```
→  exp4 ./a
Sum of  20 + 10 = 30
Sub of  20 - 10 = 30
Mul of  20 * 10 = 30
Div of  20 / 10 = 30
```

b)

```
#!/bin/bash

if (( $# == 0 )) ; then
        echo "Provide necessary command line arguments "
        echo "Usage $0 {arg1 arg2 arg3 . . . . } "
        exit
fi

echo " No. of command line arguments  = $# "
echo " Name of the executable = $0 "
echo " Arguments are : $@ "
```

output:

```
→  exp4  ./b
Provide necessary command line arguments
Usage ./b {arg1 arg2 arg3 . . . . }

→  exp4 ./b one two three
No. of command line arguments  = 3
Name of the executable = ./b
Arguments are : one two three
```

c)

```
#!/bin/bash
if [[ $# != 2 ]] ; then
        echo "Usage $0 <string1> <string2>"
        exit
fi

if [[ "$1" == "$2" ]]; then
        echo "Strings are same"
else
        echo "Strings  are not same"
fi
```

output:

```
 →  exp4  ./c
Usage ./c <string1> <string2>

 →  exp4  ./c hello hel
Strings  are not same

 →  exp4  ./c hello hello
Strings are same
```

d)

```
#!/bin/bash
if [[ $# != 1 ]] ; then
        echo "Usage <filename> "
elif [[ -f "$1" ]] ; then
        echo "Regular file "
elif [[ -d "$1" ]] ; then
        echo "Directory "
fi
```

output:

```
 →  exp4 ./d a
Regular file

 →  exp4 ./d mydir
Directory
```

e)

```
#!/bin/bash
while [ true ]
do
        echo "Ctrl + C to Exit "
        read -p "Enter 1st number : " num1
        read -p "Enter 2nd number : " num2

        echo -e "Enter the operation \n1. Add          2. Sub  3. Mul 4.Div   5. Mod "
        read -p ">> " choice
```

```
        case $choice in
                1)
                        echo "Sum =  $((num1 + num2)) "
                        ;;
                2)
                        echo "Sub =  $(( num1 - num2 )) "
                        ;;
                3)
                        echo "Mul =  $(( num1 * num2 )) "
                        ;;
                4)
                        echo "Div = $(( num1 / num2 )) "
                        ;;
                5)
                        echo "Mod = $(( num1 % num2 )) "
                        ;;
                6)
                        break ;;
                *)
                        echo "Invalid choice "
        esac
        echo
done
```

output:

```
 →  exp4   ./e
Ctrl + C to Exit
Enter 1st number : 10
Enter 2nd number : 5
Enter the operation
1. Add 2. Sub  3. Mul 4.Div   5. Mod
>> 1
Sum =  15

Ctrl + C to Exit
Enter 1st number : 10
Enter 2nd number : 5
Enter the operation
1. Add 2. Sub  3. Mul 4.Div   5. Mod
>> 2
Sub =  5

Ctrl + C to Exit
Enter 1st number : 10
Enter 2nd number : 5
Enter the operation
1. Add 2. Sub  3. Mul 4.Div   5. Mod
>> 3
Mul =  50

Ctrl + C to Exit
```

```
Enter 1st number : 10
Enter 2nd number : 5
Enter the operation
1. Add 2. Sub  3. Mul 4.Div   5. Mod
>> 4
Div = 2

Ctrl + C to Exit
Enter 1st number : 10
Enter 2nd number : 5
Enter the operation
1. Add 2. Sub  3. Mul 4.Div   5. Mod
>> 5
Mod = 0

Ctrl + C to Exit
Enter 1st number : ^C
```

f)

```
#!/bin/bash
read -p "Enter the specific character " ch

ls -A | grep "^${ch}" -c
```

output:

```
 → fd ls
f.sh  one  one.one  one.three  one.two  two  two.one  two.three

 → fd ./f
Enter the specific character : one
4

 → fd ./f
Enter the specific character : two
3
```

g)

```
#!/bin/bash
if [[ $# != 1 ]] ; then
        echo "Usage $0 <filename> "
        exit
elif [[ ! -f $1 ]] ; then
        echo "File does not exist "
        exit
fi


while read -r line ; do
echo $line | sed 's/a/A/g' | sed 's/e/E/g' | sed 's/i/I/g' | sed 's/o/O/g' | sed 's/u/U/g' |tee -a
outputfile.txt
done < "$1"
```

output:

```
→  gd  cat file.txt
a. Write a shell script program to perform arithmetic operations on two
numbers.
b. Write a shell script program demonstrate use of command line
parameters in shell script(script name, total parameters, each parameter)
c. Write a shell script program to check whether two strings sent as
command line arguments are same or not using test command.

→  gd ./g file.txt
A. WrItE A shEll scrIpt prOgrAm tO pErfOrm ArIthmEtIc OpErAtIOns On twO
nUmbErs.
b. WrItE A shEll scrIpt prOgrAm dEmOnstrAtE UsE Of cOmmAnd lInE
pArAmEtErs In shEll scrIpt(scrIpt nAmE, tOtAl pArAmEtErs, EAch pArAmEtEr)
c. WrItE A shEll scrIpt prOgrAm tO chEck whEthEr twO strIngs sEnt As
cOmmAnd lInE ArgUmEnts ArE sAmE Or nOt UsIng tEst cOmmAnd.
```

h)

```
#!/bin/bash
read -p "Enter the name of the user : " user
echo "Reversed : $(echo $user | rev)"

a=$(echo "$user"| wc -m ) && ((a--))
echo "Length   : $a"
```

output:

```
→  exp4 ./h
Enter the name of the user : sreelal
Reversed : laleers
Length   : 7
```

i)

```
#!/bin/bash

file="school.dat"
if [ -f $file ];then
        cat $file | sort -k 3 -r
fi
```

ouput:

```
→  id cat school.dat
1       sreelal 50
2       midlaj  69
3       fasal   54
4       Anas    67
5       siva    44
```

```
 →  id  ./i
2       midlaj  69
4       Anas    67
3       fasal   54
1       sreelal 50
5       siva    44
```

j)

```
#!/bin/bash
file1="file1"
file2="file2"

if [[ -f $file1 && ! -f $file2 ]] ;then
        cat $file1 > $file2
elif [[ -f $file1 && -f $file2 ]] ;then
        cat $file2 >> $file1
fi
```

output:

```
 →  jd cat file1
hello
this is file1

 →  jd  cat file2
hai
this is file2

 →  jd ./j && cat file1
hello
this is file1

hai
this is file2
```

k)

```
#!/bin/bash
read -p "Enter the no of lines : " n
for (( i=1 ; i<= n ;i++ ))  ; do
        a=""
        for (( j=0 ; j<i ; j++ )) ; do
                a="${a} ${i}"
        done
        echo $a
done
```

output:

```
→  exp4 ./k
Enter the no of lines : 5
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

l)

```
#!/bin/bash

echo "Username        : $(whoami)"
echo "Shell           : $SHELL"
echo "Home Dir        : $HOME"
echo "Os-type         : $(uname -o)"
echo "Path            : $PATH"
echo "currnt dir      : $PWD"
echo "logged usr      : $(who | wc -l)"
```

output:

```
→  exp4 ./l
Username        : hay
Shell           : /bin/bash
Home Dir        : /home/hay
Os-type         : GNU/Linux
Path            : /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/
local/games:/snap/bin
currnt dir      : /home/hay/Desktop/sreelal//os/B21CSB69/cycle1/exp4
logged usr      : 1
```

m)

```
#!/bin/bash

echo -e "OS & version , release number ,kernel version $(uname -a) \n"
echo -e "$(cat /etc/shells) \n"
echo -e "Mouse setting $(xev) \n"
echo -e "CPU information \n$(cat /proc/cpuinfo)\n"
echo -e "Memory informatin \n$(cat /proc/meminfo)\n"
echo -e "Hard Disk information\n$(sudo lshw -class disk)\n"
echo -e "Mounted\n$(lsblk)\n"
```