LAB-I

① **tty**: Displays information related to terminal.
It prints the file name of terminal connected to Std. input.

-- help displays help message and exits.
-- version prints the version information and exits.

② **Stty**: used to change and print terminal line settings.
Shows the terminal characteristics.

Stty -a / Stty -all → prints all current settings in human readable form.

③ **Script**: used to make a typescript or record all the terminal activities.
After the script command, it starts recording and printing everything including the information saved in file typescript.
Script is mainly used to capture output of command or set of command while installing a program.

④ **ls**: Lists all the names of files in a unix directory. If the command is typed with no parameters or qualifiers, the command displays files listed in current working directory

⑤ ls -l : lists the contents of the directory in a table format with columns including

(i) content permission.
(ii) no. of links to content.
(iii) owner.
(iv) group owner.
(v) size of content
(vi) modified date / time
(vii) file or directory name.

→ touch test.sh (to create a new file)
nano test.sh (opens script to type content).

In nano script type #! / bin / bash → path.
echo "hello world"

exit → Ctrl + X.

$ chmod a+x test.sh. (or) $ chmod 777
./test.sh

→ $ echo -e "Name \t USN"

→ $ echo -n "name"
(will give output on the same line)

Print 5

```
#! / bin / bash
X=5
echo $x
```

touch sum.sh.
nano sum.sh.
{}.
$ chmod a+x test.sh.
$ ./test sum.sh.

Adding two numbers.

```
#! /bin/bash.

var1=10
var2=30.
sum=$((  $var1 + $ var2))
echo $sum.
```

14/11/22.
Evaluation

Bafna Gold
Date: Page:

LAB-2

1) Area of a circle.
2) Copy a set of files to a particular directory.
use copy, read, make dir command.

①

nano area.sh

```
{   #! /bin/bash.
    echo "enter radius:"
    read r.
    echo "area is:"
    echo "3.14 * $r * $r" | bc. }
```

chmod a+x area.sh.
sh area.sh

output: enter radius:
3.
area is: 28.26

②

touch file1
touch file2.

```
{   #! /bin/bash.
    echo "enter file1"
    read x
    echo "enter file2"
    read y.
```

mkdir newdir
cp $x newdir
cp $y newdir.

cd newdir
ls.

③. create two files and compare names of flow and flower2. in first directory.

flower 1	flower 2.
Rose	Rose
Daisy	Jasmine
Marigold	Hibiscus
lily	lily
daffodil	daisy.

Nano flower 1
nano flower 2.
cmp flower1 flower2.
differ : byte 6 line 2.

diff flower 1 flower 2.

2, 3c2, 3
< daisy
< marigold.

> jasmine
> hibiscus.

5c5

< daffodil

---

daisy.

comm flower 1 flower 2.

rose

daisy.

comm: file 2 is not in sorted order.

hibiscus

lily

daisy.

marigold.

comm: file 1 not sorted

lily

dog rose

10

14/11/2022

LAB-3.

① A number is greater than 0 i.e, positive, negative or 0.

```
#! /bin/bash.

echo " enter number"
  read num
if [ $num -lt 0]
 then
 echo" number is negative"
elif [ $num -gt 0]
 then
echo " number is positive"
 else
 echo " number is zero"
 fi
```

②

```
#! /bin /bash
    echo " enter year"
    read y.
a = `expr $y %.4`
b = `expr $y %.100`
c = `expr $y %.400`

if [ $a -eq 0] && [$b -ne 0] || [$ c -eq 0]
    then
    echo "$ is leap year"
    else
    echo "$ y is not leap year".
    fi
```

① Output :

enter a number :

0

number is zero.

enter a number :

-2

number is negative

② Output :

enter year :

2024

2024 is leap year.

enter year :

2011

2011 is not a leap year.

③ Find largest number of three numbers.

```
#!/bin/bash
echo "enter number 1"
read num1
echo "enter number 2"
read num2
echo "enter number 3"
read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt num3]
then

echo "$num3 is largest"
```

```
elif [ $num2 -gt $num1 ] && [ $num3 -gt $num3)
then
echo "$num2 is greater"
else
echo "$num3 is greatest"
fi
```

Output:

```
enter number 1:
1
enter number 2:
2
enter number 3:
4
```

largest number is 4.

greatest in the command line.

10

```
#! /bin/bash.
if [$1 -gt $2 -a $1 -gt $3]
then
echo "$1 is greatest"
elif [ $2 -gt $1 -a $2 -gt $3]
then
    echo "$2 is greatest"
    elif [" $3 -gt $1 -a $3 -gt $2]
    echo "$3 is greatest".
```

19/11/12

output : sh ex.sh 2 6 7
7 is greatest.

LAB-4.

① Factorial of a number.

```
echo " enter a no."
read num
fact=1
while [ $num -gt 1 ]
do
fact=$ (( fact * num ))
num=$ (( num-1 ))
done
echo $fact.
```

Output: enter number:
3
6.

② convert fahrenheit to celcius

```
echo " enter temperature in fahrenheit "
read num
Tc=$ ( echo "scale=2; (5/9)*($num-32)" |bc )
echo " $num = $Tc "
```

Output:

enter the temperature in fahrenheit

95

95 = 34.65

③ enter salary and compute gross salary of employee (hra 20% of basic, da 10% of basic)

```
echo " enter salary."
read bs
f=$ (echo "0.2 * $bs + 0.1 * $bs + $bs " |bc)
echo = " final salary $f"
```

Output:

enter salary: 3000
final salary is 3300.

④ Sum of n natural number.

```
echo " enter number"
read n
i=1
sum=0
while [ $i -le $n]
 do
sum=$(($num + $i ))
i=$(( $i + 1))
done
echo" sum is = $sum"
```

Output:

enter size : 3
sum is : 6.

(5) find the power product.

```
echo " enter a number"
read m
echo " enter a power"
read n
exp = $ ((m**n))
echo $exp.
```

Output:

enter a number: 2

enter a power : 2

4.

(6) arthemetic operations

```
echo " enter two numbers"
read a b
echo " 1) sum"
echo " 2) diff"
echo " 3) product"
echo " 4) Quotient"
echo " 5) reminder"
echo " enter your choice"
read n.
case "$n" in
1) echo " the sum of $a and $b is `expr $a + $b` ";;
2) echo " difference is `expr $a - $b` ";;
3) echo " product is `expr $a \* $b` ";;
4) echo " Q is `expr $a / $b` " ;;
5) echo " reminder is `expr $a % $b` " ;;
```

(7) Find sum of even numbers.

```
echo "enter upper limit"
read n
$i = 2
while [ $i -lt $n ]
do
expr 'sum = $sum+$i'
expr 'i = $i + 2'
done
echo "sum is :$sum"
```

Output:

enter limit : 9

sum = 20

10

28/11/2022

LAB-5.

(8) Shell Script to print combinations of numbers 123.

```
for i in 1 2 3
do
for j in 1 2 3
do
for k in 1 2 3
do
echo $i $j $k.
done
done
done.
```

Output:

```
1 1 1
1 1 2
1 2 1
1 2 2
1 3 1
  :
3 3 1
3 3 2
3 3 3.
```

9) Shell Script to find fibonacci series up to n.

```
echo " enter the number of terms"
read n.

x=0
y=1
i=2.

echo" terms are:"
echo $x
echo $y.
```

```
while [ $i -lt $n ]
do
z=`expr $x + $y`
x=$y
y=$z
echo $z
i=`expr $i +1`
done
```

Output:

```
how many terms : 4
Terms are: 0
           1
           1
           2
```

```
how many terms : 8
terms are: 0
           1
           1
           2
           3
           5
           8
           13
```

(5) Shell Program to determine pass class.

```
echo "enter percentage"
read num
if [ $num -gt 40] && [$num -le 50]
then
echo "pass class"
elif [$num -gt 50] && [$num -lt 60]
then
echo "second class"
elif [$num -ge 60] && [$num -lt 70]
then
echo "first class"
elif [ $num -gt 70]
then
echo "distinction"
else
  echo "fail"
fi
```

10

Output:

enter percentage: 59

second class.

5/12/24

enter percentage : 99

distinction.

LAB: 6.

① Shell script to count no. of vowels in a string.

```
nano vowel.sh.

echo "enter string"
read str.
l=`expr length $str`
vowel=0.
while [ $l -gt 0 ]
do
temp=`expr $str | cut -c $l`
case $temp in

a|A) vowel=`expr $vowel+1`;;
e|E) vowel=`expr $vowel+1` ;;
i|I) vowel=`expr $vowel+1` ;;
o|O) vowel=`expr $vowel+1` ;;
u|U) vowel=`expr $vowel+1` ;;

esac
l=`expr. $l -1`
done
echo "the string has vowel $vowel"
```

Output:

```
enter string:
anik
the string has vowel 2.
```

② Shell script to check no of lines, words, characters in a file.

```
echo "enter file name"
read file
if [ -f $file ]
then
echo "file exists"
echo " no. of lines"
wc -l $file
echo " no. of char"
wc -c $file
echo " no. of words"
wc -w $file
else
echo " file doesnt exist"
fi
```

Output:

```
enter file name : anik.sh
file exists
no. of lines: 3.
no. of characters: 28
no. of words: 6.
```

anik.sh:

```
hello
my name is anikha.
hi
```

③ Shell Script to do set, shift, trap, chmod, chown, chgrp.

set -x ⟶ arguments in the same sequence as they got executed.

```
set -x
echo hello.
```

output:

```
+ echo hello.
hello.
```

Shift: Shift positional argument by value n.

```
set  1 2 3 4.
+set 1 2 3 4.
shift 2
 + shift 2.
$2
 4
```

Shell Script

```
echo " the arguments $#"
echo "  the total no. of argument $* "

echo " $ first argument is $1 "
  shift 2
echo " after shift 2 first argument is $1"
```

Output:

```
$ sh shift.sh 1 2 3
$ sh shift.sh 1 2 3
total arguments 3
the arguments are 1 2 3
first argument is 1
new first argument is 3
```

12/12/24

14/8/22.

LAB-7.

② Write a shell script for GCD and Lcm.

```
echo " enter two numbers"
read m n.
temp = ' expr $m \* $n'
while [ $m != $n ]
do
if [ $m -gt $n ]
then
  m=' expr $m - $n'
  else
  n = ' expr $n - $m'
fi
done
echo " gcd = $n"
lcm=' expr $temp /$n'
    echo " lcm = $lcm".
```

Output:

```
enter two numbers
2 4
gcd = 2
lcm = 4.
```

③

```
i,number=1
row =5
for ((i=1; i<=row; i++))
do
for ((j=1; j<=i; j++))
  do
  echo -n "$number"
  number = $ ((number + 1))
  done
```

```
number = 1
echo
done
```

Output:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

[num → ./ pyramid

(4)(a) Find all files with .html or .HTML.

① create HTML files.

nano one.html.
nano ~~two~~.html.

→ find -name "*.html"

Output:

./ one.html.
./ ~~two~~.html.

① Creating hardlink and softlink.

Hardlink

```
echo " enter directory name"
read dir
echo " enter hardlink name"
read hl
if [ -f $dir ]
  then
  echo " creating softlink "
ln $dir $hl.
    ls -ltr
    else
     echo " file doesnt exist"
     fi
```

output

enter directory name: fact.sh (already existing file)

enter hard link: an (random name).

creating softlink

total 136.

⋮

Softlink

```
echo " enter name"
read d.

echo " enter Sft link "
 read sl

 if [ -f $dir ]
   then
     echo " creating sftlink"
    ln -s
   ls -ltr
   else
   echo " file doesnt exist"
     fi
```

output:

enter name: fact.sh.

enter Sft link name: aaa.

creating sftlink

total 136.

4(b)

~~chmod~~

directories and files having 666 permission.

→ $ chmod 666. fact.sh
$ find -perm 666.

output: ./ fact.sh
./ an

→ $ chmod. 666 Downloads
$ find -perm 666

./Downloads
./ fact.sh
./ an

19/12/22