SEMESTER – V

OPERATING SYSTEMS

Practical Assignment 1 – Shell Scripts

1.1 Display the following pattern

```
#!/bin/bash
for i in $(seq 1 5)
do
str=""
for j in $(seq 1 $i)
do
str="$str$i "
done
echo "$str"
done
```

1.2 Display the following pattern.

```
#!/bin/bash
star="*"
for i in $(seq 1 5)
do
str=""
```

```
for j in $(seq 1 $i)

do

str="$str$star "

done

echo "$str"

Done
```

1.3 Display the following pattern.

```
#!/bin/bash

for j in $(seq 1 5)

do

echo ""

for i in $(seq 1 $((5-j)))

do echo -n " "

done

for n in $(seq 1 $j)

do

echo -n "$j "

done

done

echo ""
```

1.4 Display the following pattern.

```
#!/bin/bash


for j in $(seq 1 5)

do

echo ""


for i in $(seq 1 $((5-j)))

do echo -n " "

done


for n in $(seq 1 $j)

do

echo -n "$j "

done

done


for j in $(seq 4 -1 1)

do

echo ""


for i in $(seq 1 $((5-j)))

do echo -n " "
```

```
done


for n in $(seq 1 $j)
do
echo -n "$j "
done
done
echo ""
```

1.5 Display the following pattern.

```
#! /bin/bash


for((i=1;i<=5;i++))
do
    for((j=1;j<=5-i;j++))
    do
            echo -n bash' '
    done
    for((k=1;k<=i;k++))
    do
            echo -n '*'
            echo -n ' '
    done
    echo
```

```
done
for((i=5-1;i>=1;i--))
do
    for((j=1;j<=5-i;j++))
    do
            echo -n ' '
    done
    for((k=1;k<=i;k++))
    do
            echo -n '*'
            echo -n ' '
    done
    echo
done
```

1.6 Display the following pattern.

```
#! /bin/bash


for((i=1;i<=5;i++))
do
    for((j=1;j<=5-i;j++))
    do
            echo -n ' '
    done
```

```
    for((k=1;k<=i;k++))
    do
            echo -n $i
            echo -n ' '
    done
    echo
done
```

2. Write a shell script to find the factorial of a given no.

```
#! /bin/bash

echo -n "Enter Number : "
read N
for((i=N-1;i>=1;i--))
do
    ((N=N*i))
done
echo -n "Ans : "
echo $N
```

3. Write a shell script to find the largest of three numbers and also find the total and average.

```
#! /bin/bash
echo enter three number
```

```
read a
read b
read c
if [ $a -gt $b ]
then
        if [ $a -gt $c ]
        then
        echo Max is $a
        else
        echo Max is $c
        fi
else
        if [ $b -gt $c ]
        then
        echo Max is $b
        else
        echo Max is $c
        fi
fi
```

5. Write a shell script to find whether a given year (4 digits) is leap year or not.

```
#!/bin/bash

echo -n "Enter year : "

read year

if [ $((year%4)) -eq 0 ]

then

    if [ $((year%100)) -eq 0 ]

    then

            if [ $((year%400)) -eq 0 ]

            then echo "Leap Year"

            else echo "Not a Leap Year"

            fi

    else echo "Leap Year"

    fi

else echo "Not a Leap Year"

fi
```

Output :

(base) [vatsal@localhost OS LAB]$ sh leapyear.sh

Enter year : 1700

Not a Leap Year

(base) [vatsal@localhost OS LAB]$ sh leapyear.sh

Enter year : 2020

Leap Year

6. Write a shell script to find the sum of first n numbers.

```
#!/bin/bash

echo -n "Enter value of n : "

read n

ans=$(((n*(n+1))/2))

echo $ans
```

7. Write a shell script to check whether a given no. is prime or not.

```
#!/bin/bash

echo "Enter a number : "

read a

b=$((a-1))

flag=1

for i in $(seq 2 "$b")
```

```
do
#echo "$((a%i))"
if [ $((a%i)) -eq 0 ]
then flag=$((flag-1));break
fi
done
#echo "$flag"
if [ "$flag" -eq 1 ]
then echo "Number is prime"
else echo "Number is not prime"
fi
```

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

```
#! /bin/bash
echo -n "Enter Number : "
read N
for((i=1;i<=10;i++))
do
    echo $N '*' $i '=' $((N*1))
done
```

9. Write a command file that displays the following:

a) Calendar of the current month and year.

b) Current date in dd/mm/yy and time.

c) Display “Good Morning / Good Afternoon / Good Evening” according to the current login time.

d) User name, user’s home directory.

e) Terminal name, terminal type.

f) Machine name.

g) No. of user currently logged in.

```
#!/bin/bash

echo "Calender : ";cal

echo "Date : ";date +"%d/%m/%y";echo ""

echo "Time : ";date +"%T";echo ""

check=`date +%H`

if [ $check -gt 6 -a $check -lt 12 ]

then echo "Good Morning"

elif [ $check -ge 12 -a $check -lt 18 ]

then echo "Good Afternoon"
```

```
else echo "Good Evening"

fi

echo ""

echo "User name : ";whoami;echo ""

echo "Home directory : ";echo $HOME;echo ""

echo "Terminal name : ";tty;echo ""

echo "Terminal Type : ";echo $TERM;echo ""

echo "Machine name : ";uname -a;echo ""

echo "Number of users currently logged in : ";users | wc -w

# Output :

Calender :

          August 2019

Su Mo Tu We Th Fr Sa

          1  2  3

 4  5  6  7  8  9 10

11 12 13 14 15 16 17

18 19 20 21 22 23 24

25 26 27 28 29 30 31

Date :

17/08/19

Time :

21:54:38

Good Evening

User name :
```

vatsal

Home directory :

/home/vatsal

Terminal name :

/dev/pts/0

Terminal Type :

xterm-256color

Machine name :

Linux mr5-VirtualBox 4.15.0-46-generic #49-Ubuntu SMP Wed Feb 6 09:33:07 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux

Number of users currently logged in :

1

10. Write a shell script to find the sum of n numbers which are passed by command line argument.

```
#!/bin/bash

ans=0

for i in $@

do

ans=$((ans+i))

done

echo $ans

# Output

:~/Desktop/Operating System$ sh sumofnCommandline.sh 2 3 4
```

9

:~/Desktop/Operating System$ sh sumofnCommandline.sh 2 3 4 7 10

26

11. Write a shell script to find the sum of digits of a number entered through command line argument and find whether sum is even or not.

```
 #!/bin/bash
n=$1
ans=0
while [ $n -ne 0 ]
do
r=$((n%10))
ans=$((ans+r))
n=$((n/10))
done
if [ $((ans%2)) -eq 0 ]; then
echo "Even"
else echo "Odd"
Fi
```

~/Desktop/Operating System$ sh sumofdigitsCommandline.sh 111

Odd

~/Desktop/Operating System$ sh sumofdigitsCommandline.sh 1111

Even

:~/Desktop/Operating System$ sh sumofdigitsCommandline.sh 1221

Even

~/Desktop/Operating System$ sh sumofdigitsCommandline.sh 12211

Odd

~/Desktop/Operating System$ sh sumofdigitsCommandline.sh 122111

Even

'

12. Write a shell script to print all the values which are passed by command line argument in reverse way. If total values entered through command line argument are more than 5 print “Invalid number of arguments”.

```
#!/bin/bash


if [ $# -gt 5 ];then

echo "invalid number of Arguments"

else

n=$#

args="$@"

echo $args

for i in $(seq $((n-1)) -1 0)
```

```
do
echo $i
#echo -n "${args[$i]}"
done
fi
echo ""
```

13. Write a shell script to check whether a given user is currently logged in or not.

```
#!/bin/bash
echo -n "Enter name of the user : "
read name
Logged=`users`
for user in $Logged
do
if [ $user = $name ];then
echo "User is Logged in currently"
exit
fi
done
echo "User not logged in currently"
#Output :
(base) [vatsal@localhost OS LAB]$ sh userLoggedin.sh
Enter name of the user : vatsal
```

User is Logged in currently

(base) [vatsal@localhost OS LAB]$ sh userLoggedin.sh

Enter name of the user : ttt

User not logged in currently

'

16. Write a shell script to remove all the zero sized files from the current directory.

```
#!/bin/bash


echo -n "Enter name of the directory :"
read dire
if [ ! -d "$dire" ]
then
echo "Directory does not exist"
else
for i in `find $directory -type f -size 0`
do
rm -i $i
done
Fi
```

17. Combine Emp1 and Emp2 in file Emp3 horizontally and vertically.

```
#!/bin/bash

IFS=

echo "File one : "

echo `cat $1`

echo "File two : "

echo `cat $2`

`paste $1 $2 > Horizontal`

echo "Horizontal : "

echo `cat Horizontal`

echo "Vertical : "

`cat $1 $2 > Vertical`

echo `cat Vertical`

: '

cp25@cp25-OptiPlex-3050:~/Desktop/17BIT028$ sh 17.sh one two

File one :

a

a

a

a

File two :

b
```

b

b

b

Horizontal :

a b

a b

a b

a b

Vertical :

a

a

a

a

b

b

b

b

'

22. Write a shell script to change the suffix of all your *.txt files to .dat.

```
#!/bin/bash


for file in *.txt

do

mv -- "$file" "${file%.txt}.text"

done
```

palindrome number

```
#! /bin/bash

echo -n "Enter Number : "

read N

num=$N

c=''

while [ $N -ne 0 ]

do

    ll=$((N%10))

    c=$c$ll

    N=$((N/10))

done

if [ $c -eq $num ]
```

```
then
    echo "yes"
else
    echo "No"
fi
```