

VISHWAKARMA INSTITUTE OF TECHNOLOGY

COMPUTER ENGINEERING

Name: Ayush Surendra Billade

Division: CSA

Roll Number: 38

Subject: Operating System (OS) LAB

Shell Script

Write a shell script for

1.For Calculator using command line arguments

```
#!/bin/bash
```

```
if [ $# -ne 3 ]
```

```
t
```

```
then
```

```
    echo "Usage: $0 num1 operator num2"
```

```
    exit 1
```

```
fi
```

```
num1=$1
```

```
operator=$2
```

```
num2=$3
```

```
case $operator in
```

```
    +)
```

```
        result=$((num1 + num2))
```

```
    ;;
```

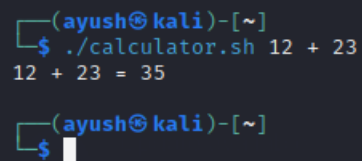
```

-)
    result=$((num1 - num2))
    ;;
\*)
    result=$((num1 * num2))
    ;;
/)
    result=$((num1 / num2))
    ;;
*)
    echo "Invalid operator: $operator"
    exit 1
    ;;
esac

echo "$num1 $operator $num2 = $result"

```

Output:



```

(ayush@kali)~$ ./calculator.sh 12 + 23
12 + 23 = 35
(ayush@kali)~$

```

2. To accept the strings & to reverse the string.

Code: #!/bin/bash

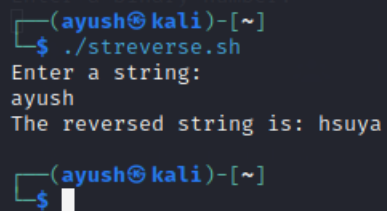
```

echo "Enter a string: "
read str
len=${#str}

```

```
rev=""  
  
for (( i=$len-1; i>=0; i-- ))  
do  
    rev="$rev${str:$i:1}"  
done  
  
echo "The reversed string is: $rev"
```

Output:



```
(ayush@kali)-[~]  
$ ./streverse.sh  
Enter a string:  
ayush  
The reversed string is: hsuya  
  
(ayush@kali)-[~]  
$
```

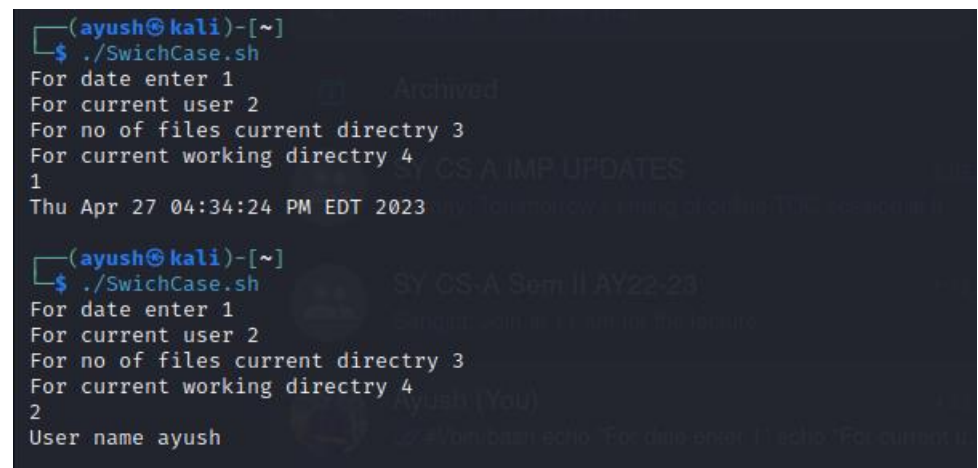
3. To execute linux commands using case statement.

Code:

```
#!/bin/bash  
  
echo "For date enter 1"  
echo "For current user 2"  
echo "For no of files current directry 3"  
echo "For current working directry 4"  
  
read n  
  
case $n in  
    1)  
        echo `date`  
        ;;  
    2)  
        u="$USER"  
        echo "User name $u"
```

```
;;  
3)  
echo "$(ls | wc -l)"  
echo "$(ls)"  
;;  
4)  
echo "$(pwd)"  
;;  
*)  
echo "Invalid operator: $operator"  
exit 1  
;;  
esac
```

Output:



The screenshot shows two terminal sessions. The first session shows the script being run with inputs 1, 2, 3, and 4, resulting in the number of files, the directory listing, and the current directory being printed. The second session shows the script being run with inputs 1, 2, 3, and 4, resulting in the number of files, the directory listing, the current directory, and the user name being printed.

```
(ayush@kali)-[~]  
$ ./SwichCase.sh  
For date enter 1  
For current user 2  
For no of files current directry 3  
For current working directry 4  
1  
Thu Apr 27 04:34:24 PM EDT 2023  
  
(ayush@kali)-[~]  
$ ./SwichCase.sh  
For date enter 1  
For current user 2  
For no of files current directry 3  
For current working directry 4  
2  
User name ayush
```

```
(ayush@kali)-[~]
$ ./SwichCase.sh
For date enter 1
For current user 2
For no of files current directry 3
For current working directry 4
3
25
arraysort.sh
binary_decimal.sh
calculator.sh
coa
COA Assignment
Desktop
DOCS
Documents
Downloads
d.txt
factorial.sh
file1
hello.sh
hellow.sh
ip.txt
Music
Pictures
program.sh
Public
pyramid.sh
streverse.sh
stringreverse.sh
SwichCase.sh
Templates
Videos

(ayush@kali)-[~]
$ ./SwichCase.sh
For date enter 1
For current user 2
For no of files current directry 3
For current working directry 4
4
/home/ayush
```

4. To print the pyramid of *

Code:

```
#!/bin/bash

echo "Enter the number of rows for the pyramid: "

read rows

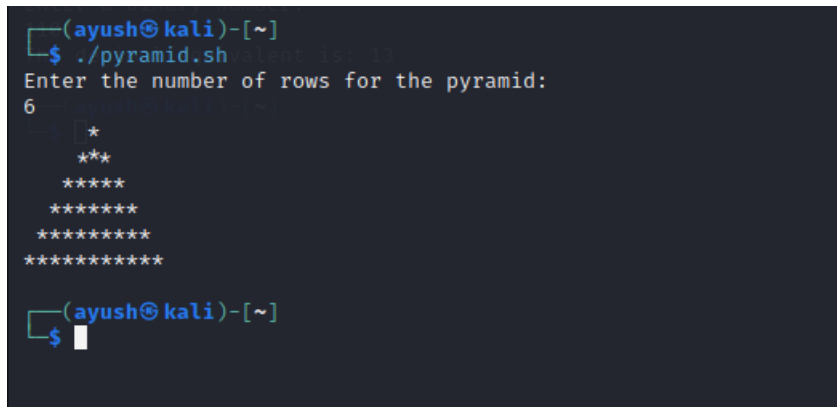
for (( i=1; i<=rows; i++ ))
do
    for (( j=1; j<=rows-i; j++ ))
    do
```

```

        echo -n " "
done+++++
for (( k=1; k<=2*i-1; k++ ))
do
    echo -n "*"
done
echo
done

```

Output:



```

(ayush@kali)-[~]
$ ./pyramid.sh
Enter the number of rows for the pyramid:
6
  *
 ***
*****
*****
*****
*****
*****
(ayush@kali)-[~]
$

```

5.To write a function for factorial of a number

Code:

```
#!/bin/bash
```

```

factorial() {
    if [ $1 -eq 0 ]
    then
        echo 1
    else
        prev=$(factorial $(( $1 - 1 )))
        echo $(( $1 * $prev ))
    fi
}

```

```
}
```

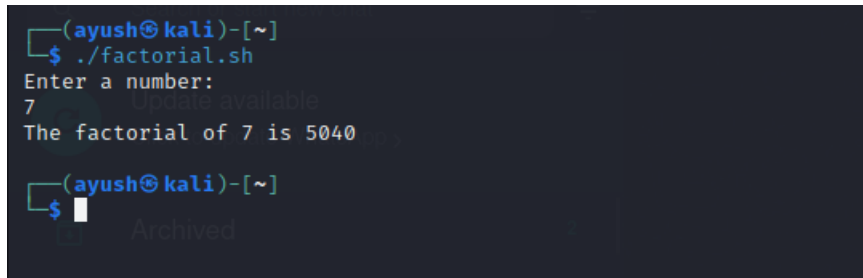
```
echo "Enter a number: "
```

```
read num
```

```
fact=$(factorial $num)
```

```
echo "The factorial of $num is $fact"
```

Output:

A terminal window with a dark background. The prompt is (ayush@kali)-[~]. The user enters \$./factorial.sh. The script prompts "Enter a number:" and the user enters 7. The script outputs "The factorial of 7 is 5040". The prompt returns to (ayush@kali)-[~]. There is a faint "Update available" notification in the background.

```
(ayush@kali)-[~]  
$ ./factorial.sh  
Enter a number:  
7  
The factorial of 7 is 5040  
  
(ayush@kali)-[~]  
$
```

6. To sort the given elements using any sorting method.

```
#!/bin/bash
```

```
echo "Enter number of elements"
```

```
read n
```

```
echo "enter Numbers in array:"
```

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

```
do
```

```
    read nos[$i]
```

```
done
```

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

```
do
```

```
    for (( j = $i; j < $n; j++ ))
```

```
do
    if [ ${nos[$i]} -gt ${nos[$j]} ];
    then
        t=${nos[$i]}
        nos[$i]=${nos[$j]}
        nos[$j]=$t
    fi
done
done
echo -e "\nSorted Numbers "
for (( i=0; i < $n; i++ ))
do
    echo ${nos[$i]}
done
```

Output:

```
(ayush@kali)-[~]
$ ./arraysort.sh
Enter number of elements
5
enter Numbers in array:
45
56
24
15
35

Sorted Numbers
15
24
35
45
56
```