SHELL SCRIPTING:

A file containing various commands to be executes on the terminal.

GENERAL VIM COMMANDS:

VIM: INSERT MODE Press I

To save and exit escape; colon; w;q.

Cat /etc/shells

SAMPLE PROGRAMS:

```
#!/bin/bash
        echo "Hello WOrld"
        read name
        echo "$name"
        name1= "abhay"
        echo "$name1"
```

```
#!/bin/bash
    echo "Hello WOrld"
        c=100
        if(($c>90))
                then
                        echo "Greater than 90"
                fi
        n=1
        while(($n<=10))
                do
                        echo $n
                        n=$((n+1))
                done
        for((i=0;i<5;i++))
                do
                        echo $i
                done
        read name

        case $name in
                "A") echo "welcome a";;
                "B") echo "welcome b";;
                *) echo "wrong option";;
        esac
```

```
#!/bin/bash
	echo "Hello WOrld"
	num1 =4
	num2 =2
	echo $((num1+num2))
	    echo $((num1-num2))
	    echo $((num1*num2))
	    echo $((num1%num2))
```

PROGRAMS:

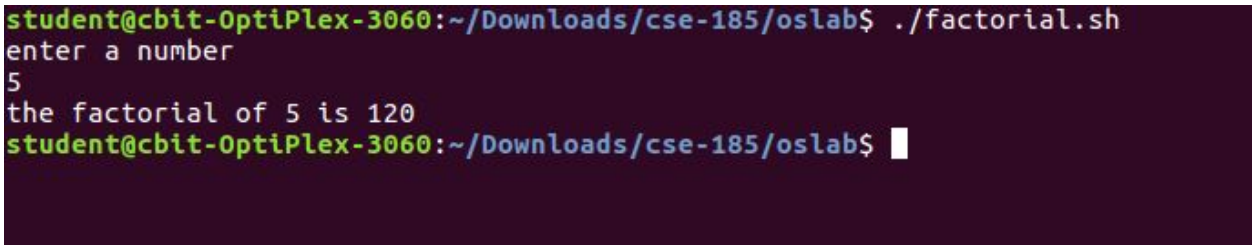
1)
AIM: Shell Script to show factorial of given number

DESCRIPTION: Printing factorial using a loop in shell script.

PROGRAM:

```
#!/bin/bash
echo "Enter numbr to find factorial"
read num
ans=1
for((i=1;i<=$num;i++))
	do
		ans=$((ans*i))
	done
	echo $ans
```

OUTPUT:



2)
AIM: Shell Script to whether given number is odd or even.

DESCRIPTION: Showing if odd or even by condition statements.

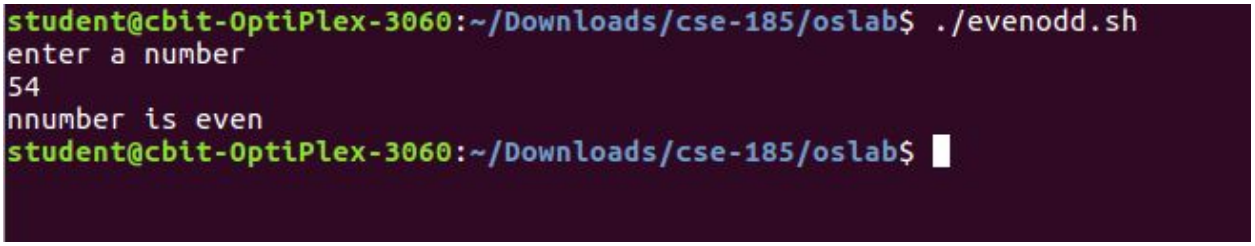
PROGRAM:

```
#!/bin/bash

read num

if (($num%2==0))
then
        echo "ITS Even"
else
        echo "ITS Odd"
fi
```

OUTPUT:



3)
AIM: Shell Script to show reverse of given number

DESCRIPTION: Printing reverse of a given number using pipe.

PROGRAM:

```
#!/bin/bash

        echo "Entr number to be reversed"
        read num
    echo $num | rev
```

OUTPUT:

```
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$ ./reverse.sh
enter a number
7654231
reverse is 1324567
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$
```

4)

AIM: Shell Script to show sum of digits of given number.

DESCRIPTION: Printing sum of digits by adding individually.

PROGRAM:

```
#! /bin/bash
echo "enter a number"
read n
res=0
while (($n!=0))
do
        res=$((res+(n%10)))
        n=$((n/10))
done
echo "sum of digits is $res"
```

OUTPUT:

```
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$ ./sumofdig.sh
enter a number
543
sum of digits is 12
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$
```

5)

AIM: Shell Script to determine given number is palindrome or not.

DESCRIPTION: Determine given number is palindrome or not using conditional statements.

PROGRAM:

```
#! /bin/bash
echo "enter a number"
read n
res=0
temp=$n
while (($n!=0))
do
	res=$(((res*10)+(n%10)))
	n=$((n/10))
done
if (($temp==$res))
then
		echo "number is palindrome"
else
		echo "number is not palindrome"
fi
```

OUTPUT:

```
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$ ./palin.sh
enter a number
68321
number is not palindrome
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$ ./palin.sh
enter a number
12321
number is palindrome
```

6)

AIM: Shell Script to determine given number is armstrong or not.

DESCRIPTION: Determine given number is armstrong or not using conditional statements.

PROGRAM:

```
#! /bin/bash
echo "enter a number"
read n
res=0
r=0
temp=$n
while (($n!=0))
do
        r=n%10
        res=$((res+(r*r*r)))
    n=$((n/10))
done
if (($temp==$res))
then
        echo "number is armstrong"
else
        echo "number is not armstrong"
fi
```

OUTPUT:

```
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$ ./arm.sh
enter a number
54
number is not armstrong
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$ ./arm.sh
enter a number
153
number is armstrong
```

7)

AIM: Shell Script to do calculations.

DESCRIPTION: Do calculation using conditional statements with different arithmetic operators.

PROGRAM:

```
#! /bin/bash
echo "enter number 1"
read n1
echo "enter number 2"
read n2
echo "enter the operator"
read op
case $op in
        "+")
                echo "sum is $((n1+n2))";;
        "-")
                echo "diff is $((n1-n2))";;
        "*")
          echo "product is $((n1*n2))";;
      "/")
          echo "division is $((n1/n2))";;
        "%")
          echo "mod is $((n1%n2))";;

        *)
                echo "wrong choice";;
esac
```

OUTPUT:

```
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$ ./arithcalc.sh
enter number 1
45
enter number 2
32
enter the operator
1
wrong choice
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$ ./arithcalc.sh
enter number 1
4
enter number 2
6
enter the operator
-
diff is -2
```

8)

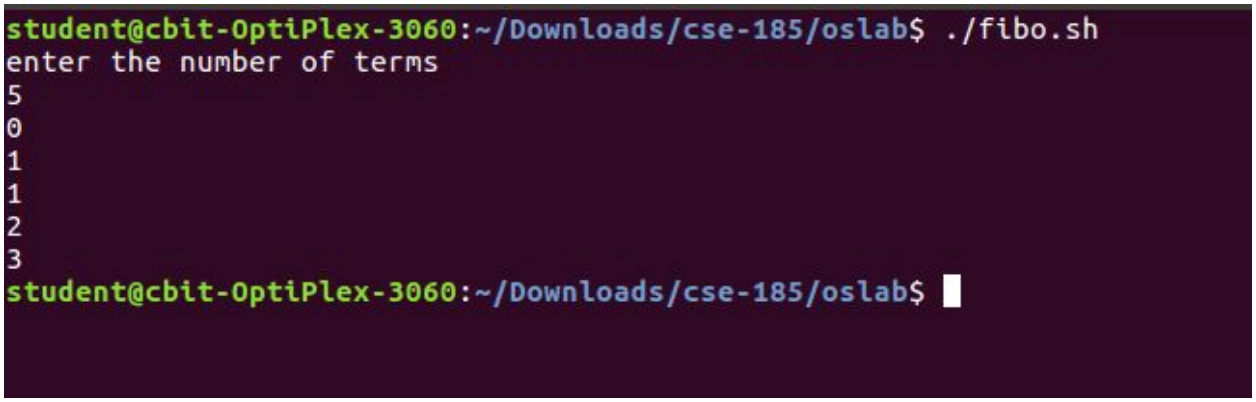
AIM: Shell Script to print fibonacci series.

DESCRIPTION: Printing fibonacci series using loops.

PROGRAM:

```
echo "enter the number of terms"
read n
f1=0
f2=1
echo $f1
echo $f2
for((i=0;i<$n-2;i++))
do
        temp=$((f1+f2))
        echo $temp
        f1=$f2
        f2=$temp
Done
```

OUTPUT:



9)

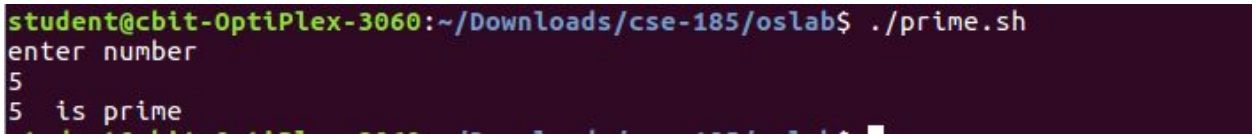
AIM: Shell Script to determine given number is prime or not.

DESCRIPTION: Determine given number is prime or not using conditional statements.

PROGRAM:

```
#!/bin/bash
echo "enter number"
read no
f=0
if((no==0 || no==1))
then
        echo $no " not prime"
else
for((i=2;i<$no;i++))
do
        if(($((no%i))==0))
        then f=1
                break
        fi
done
if((f==0))
then
        echo $no " is prime"
else
        echo $no " not prime"
fi
fi
```

OUTPUT:



10)

AIM: Shell Script to show square and cube power of given number.

DESCRIPTION: Using multiplication operator printing square and cube of a given number.

PROGRAM:

```
#! /bin/bash
echo "enter a number"
read n
echo "square of a number is $((n*n))"
echo "cube of a number is $((n*n*n))"
```

OUTPUT:

```
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$ ./sqcubge.sh
enter a number
4
square of a number is 16
cube of a number is 64
student@cbit-OptiPlex-3060:~/Downloads/cse-185/oslab$
```

11)

AIM: Shell Script to determine given number is perfect or not.

DESCRIPTION: Determine given number is perfect or not using conditional statements.

PROGRAM:

```
#! /bin/bash
echo "enter a number"
read n
sum=0
for((i=1;i<=$n/2;i++))
do
        z=$((n%i))
        if (($z==0))
        then
                sum=$((sum+i))
        fi
done
if (($sum==$n))
```

```
then
      echo "it is perfect number"
else
          echo "it is not a perfect number"
fi
```

OUTPUT: