**Name:Sangwe Gopal Bhagwanrao**

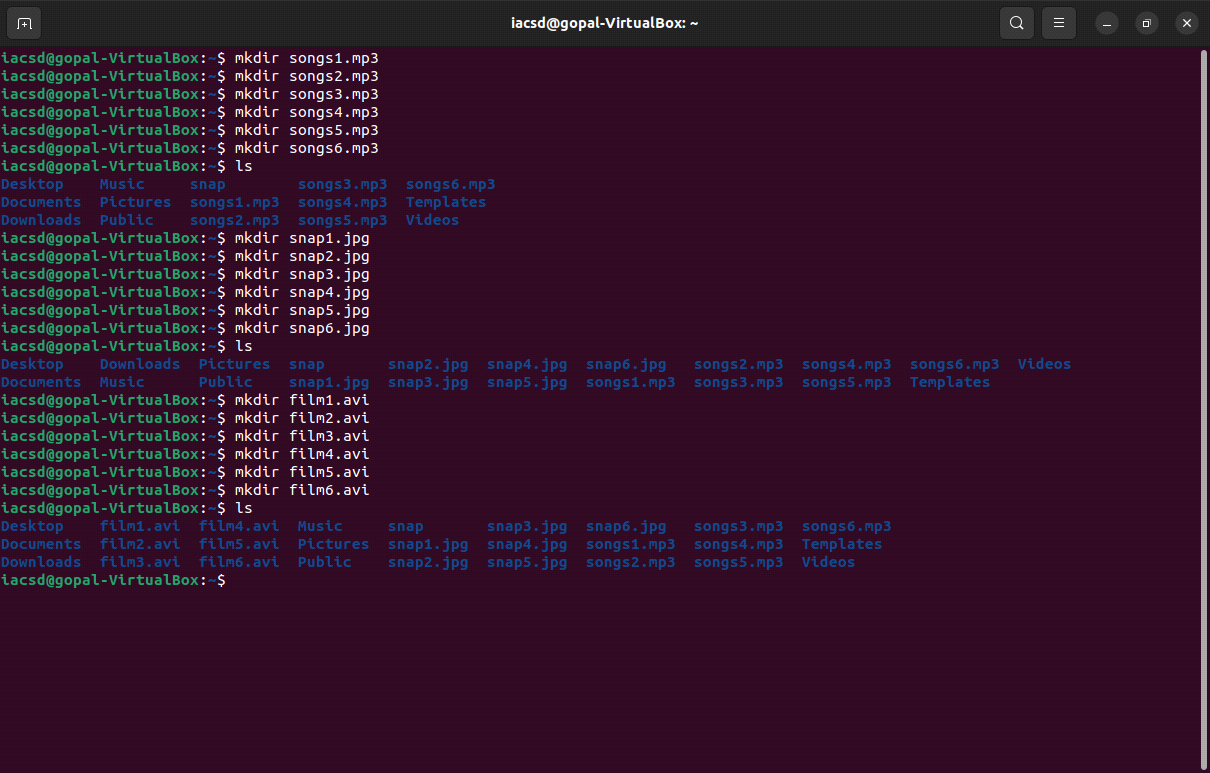
**Roll no:239083**

**ASS -1**

1. In your home directory, create sets of empty practice files

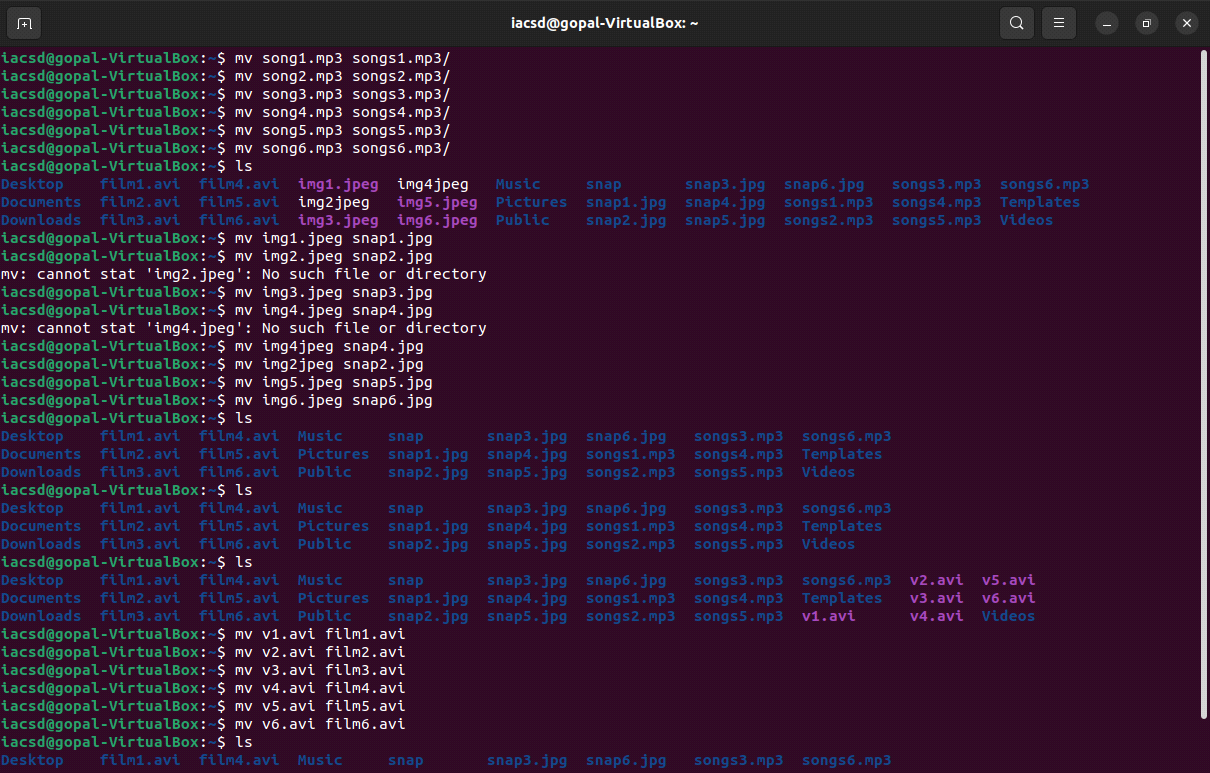
* Create 6 files with names of the form songsX.mp3.
* Create 6 files with names of the form snapX.jpg.
* Create 6 files with names of the form filmX.avi.

In each set, replace X with the numbers 1 through 6.

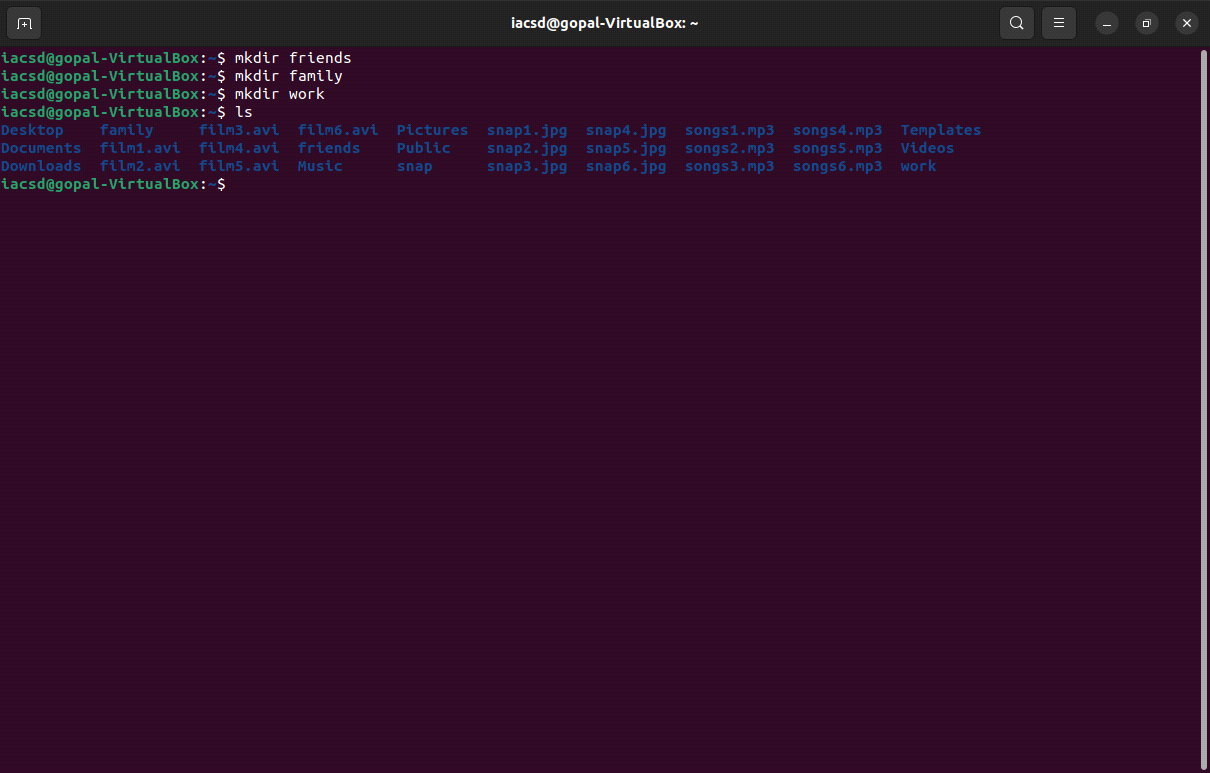


2. From your home directory,

* Move songs file into your Music subdirectory.
* Move snap file into your Pictures subdirectory.
* Move your movie files into Videos subdirectory



3. Create 3 subdirectories for organizing your files named friends,family,work



4. Copy files (all types ) containing numbers 1 and 2 to the friends folder.

Copy files (all types) containing numbers 3 and 4 to the family folder.

Copy files (all types) containing numbers 5 and 6 to the work folder.

#!/bin/bash

# Define source directory and destination folders

source\_dir="/home/iacsd"

friends\_folder="/home/iacsd/friends"

family\_folder="/home/iacsd/family"

work\_folder="/home/iacsd/work"

# Copy files containing numbers 1 and 2 to the friends folder

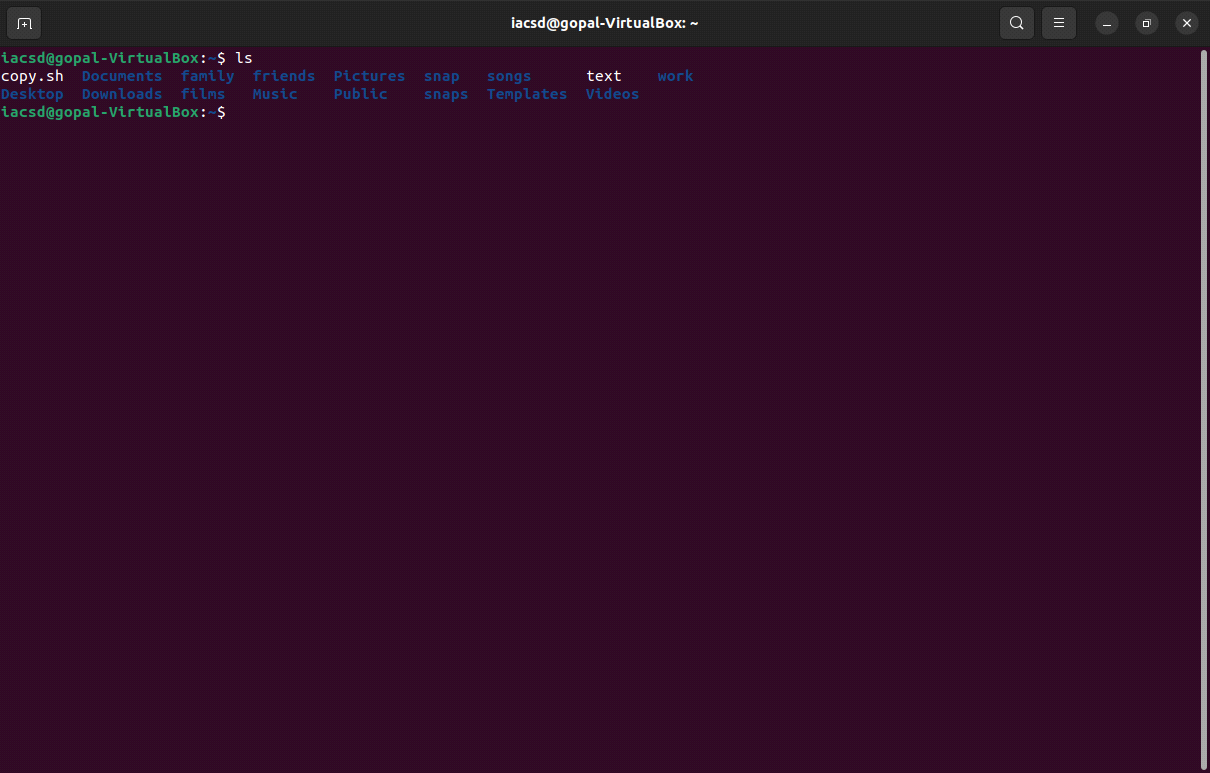
find "$source\_dir" -type f -exec grep -q '[12]' {} \; -exec cp {} "$friends\_folder" \;

# Copy files containing numbers 3 and 4 to the family folder

find "$source\_dir" -type f -exec grep -q '[34]' {} \; -exec cp {} "$family\_folder" \;

# Copy files containing numbers 5 and 6 to the work folder

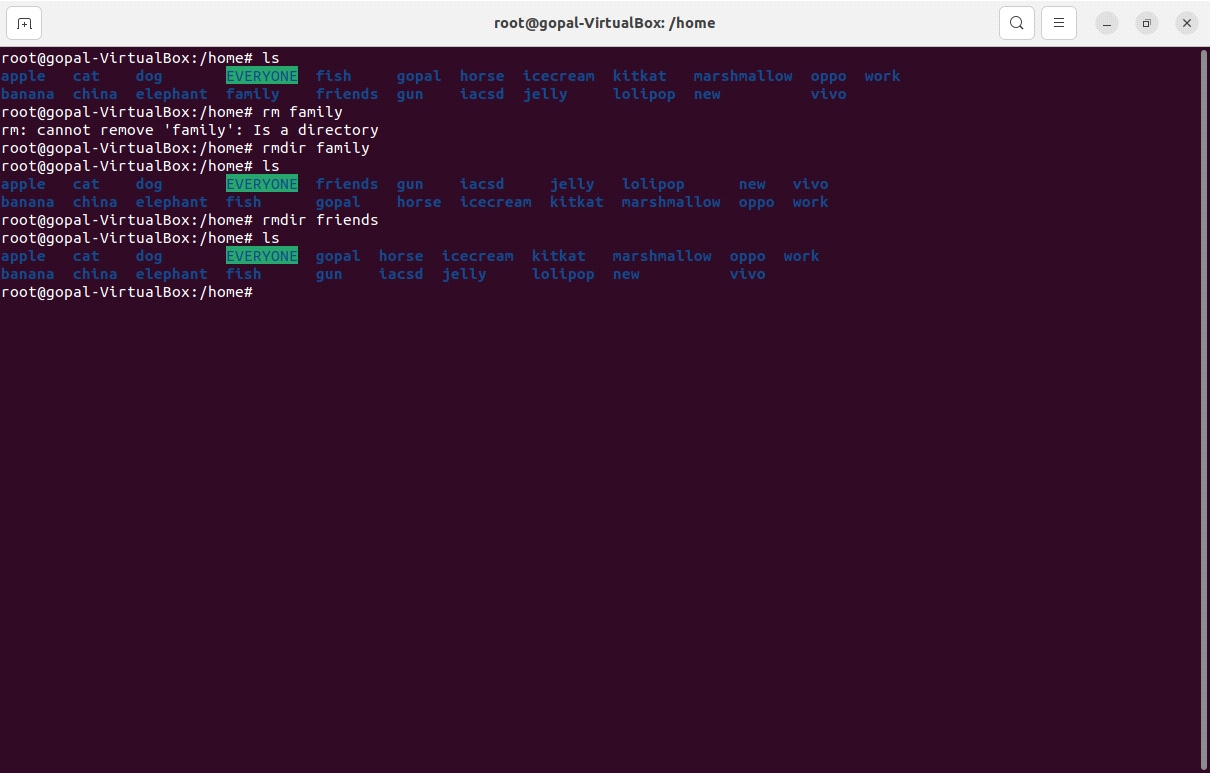
find "$source\_dir" -type f -exec grep -q '[56]' {} \; -exec cp {} "$work\_folder" \;



**ASS-2**

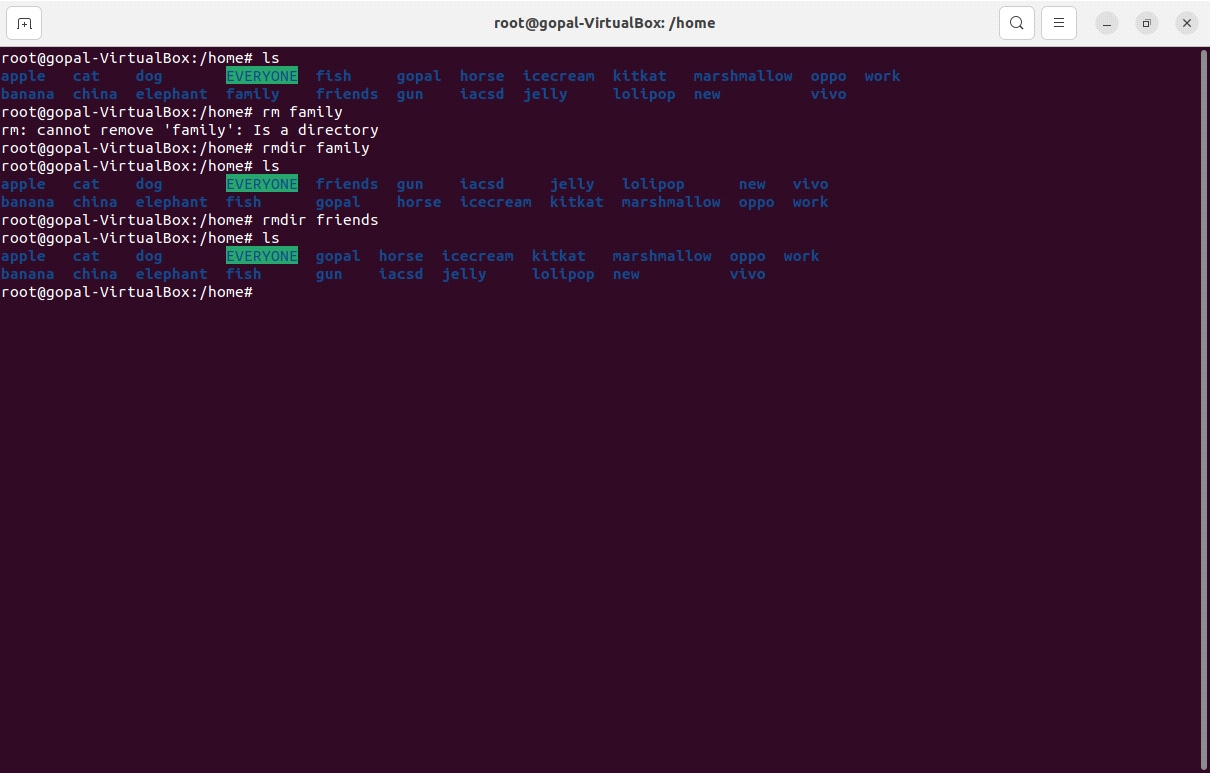
6. Delete all files in family subdirectory.

**rm Music/family/\* Pictures/family/\* Videos/family/\***



7. Delete friends subdirectory

**rm -r Music/friends Pictures/friends Videos/friends**



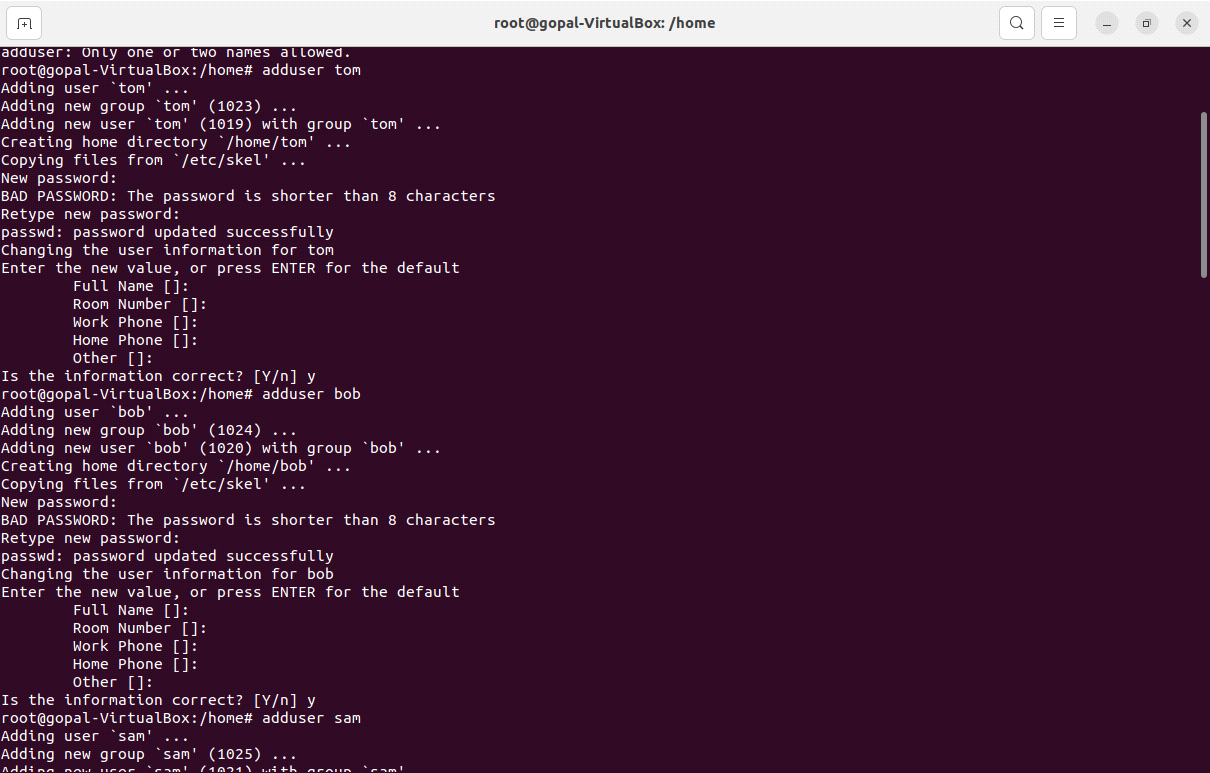
8. Create user tom , bob , sam , prince

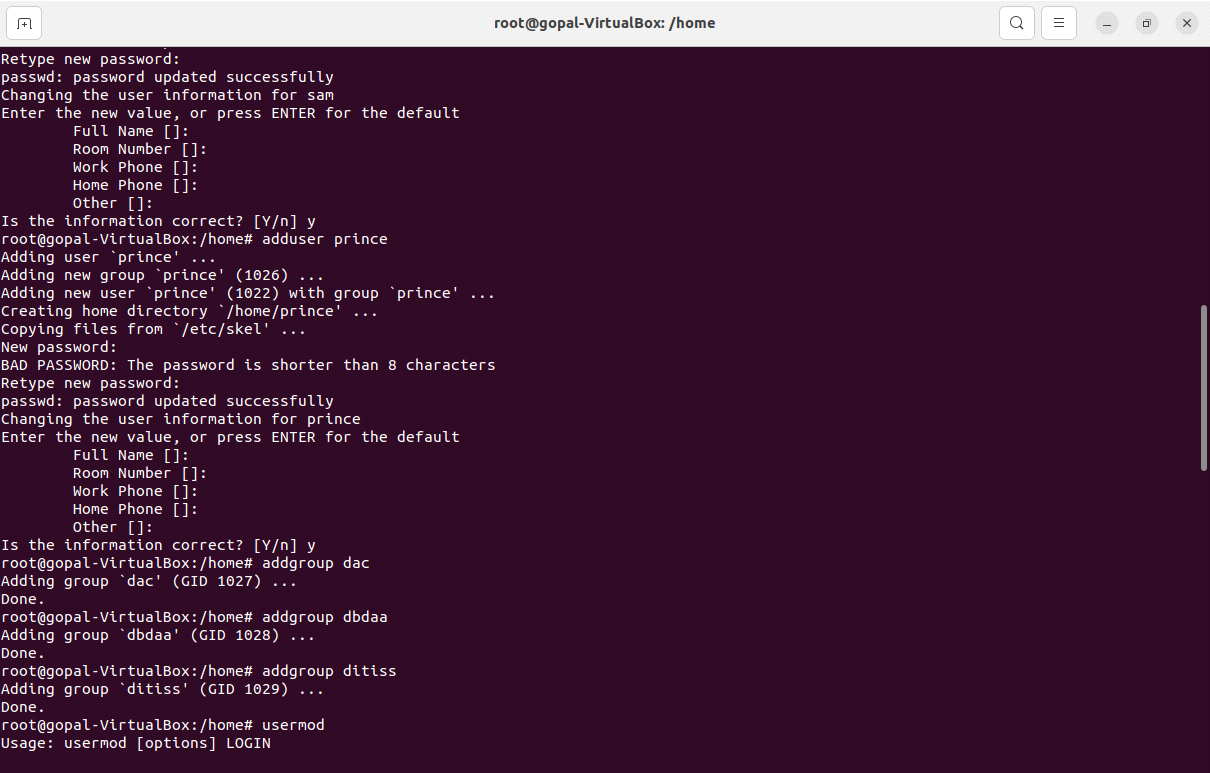
**adduser tom**

**adduser bob**

**adduser sam**

**adduser prince**





9. Create Group dac , dbda ,ditiss

**addgroup dac**

**addgroup dbda**

**addgroup ditiss**

10. add user

Tom in dac

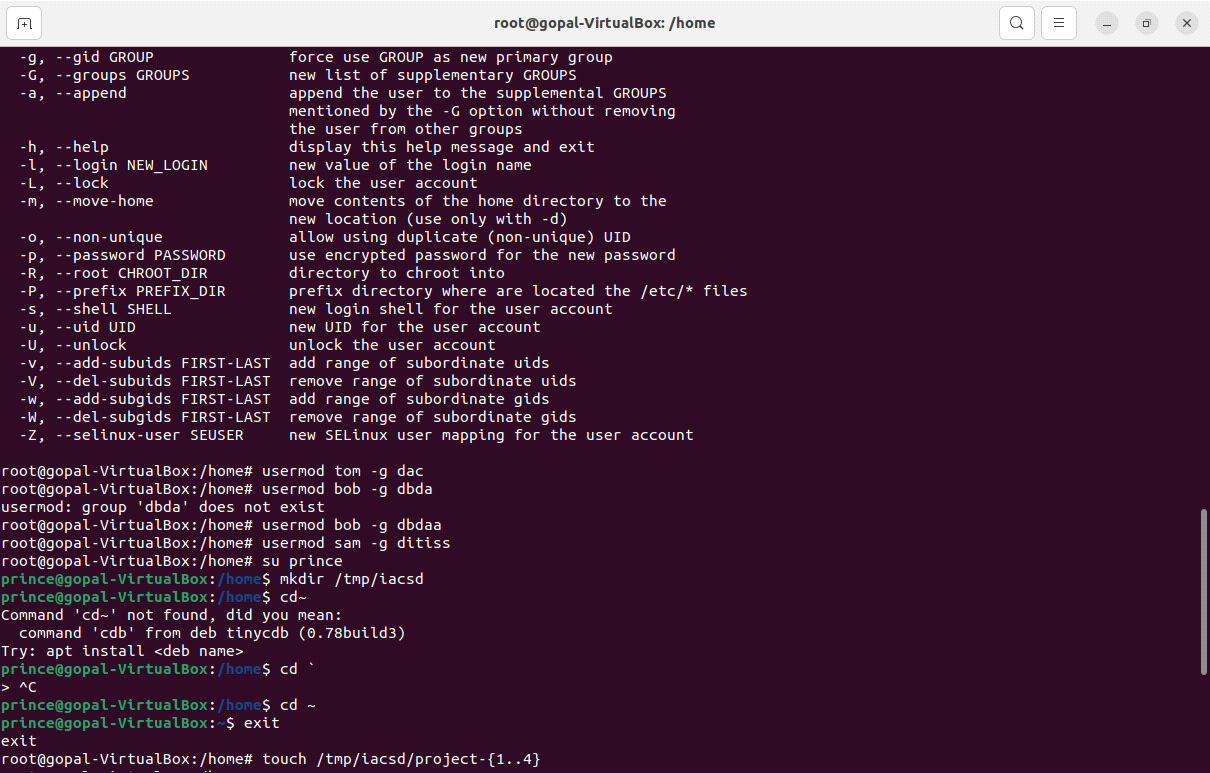
Bob in dbda

Sam in ditiss

**usermod tom -g dac**

**usermod bob -g dbda**

**usermod sam -g ditiss**



11. login as prince and create iacsd directory in /tmp and create 4 files in iacsd

with name project-1 project-2 upto 4

**su prince**

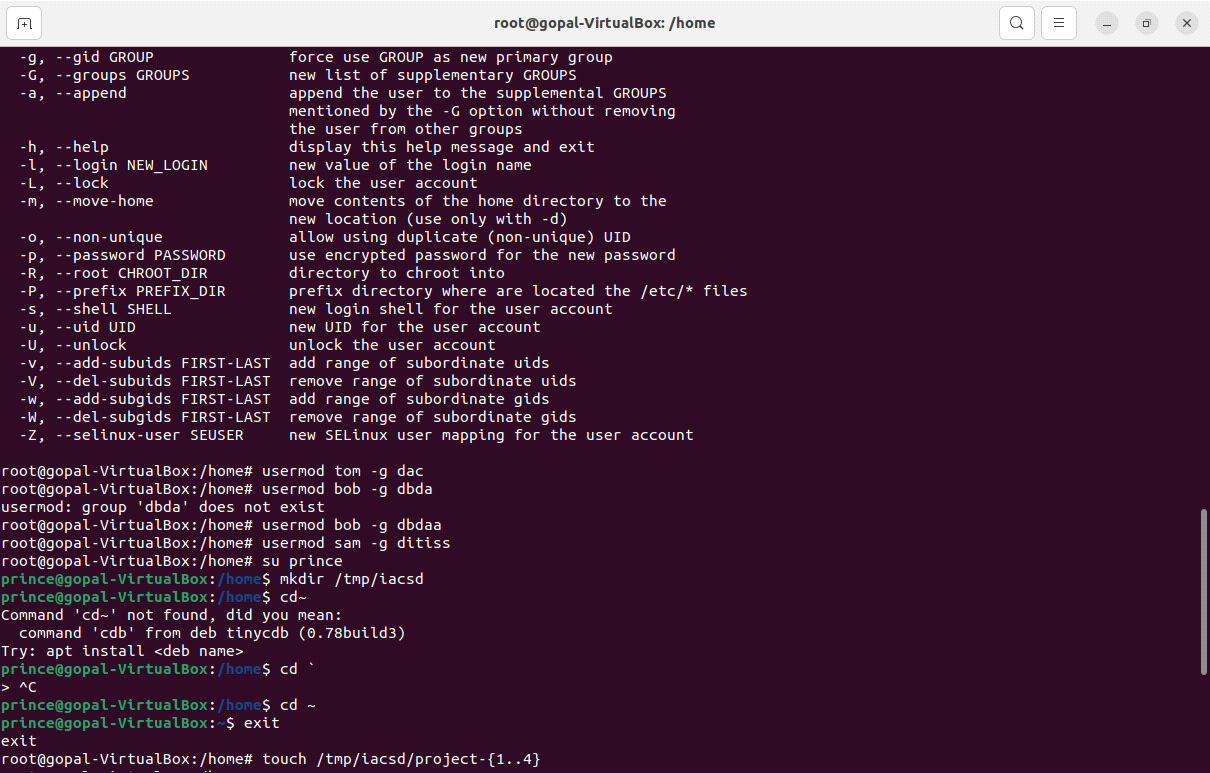
**mkdir /tmp/iacsd**

**cd ~**

**exit**

**sudo -s**

**touch /tmp/iacsd/project-{1..4}**



12. assign permissions to project files as below

Project-1 – tom should be owner of this

Project-2 – dac should be owner of this

Project-3 --- others should not have any permission but tom should have rw

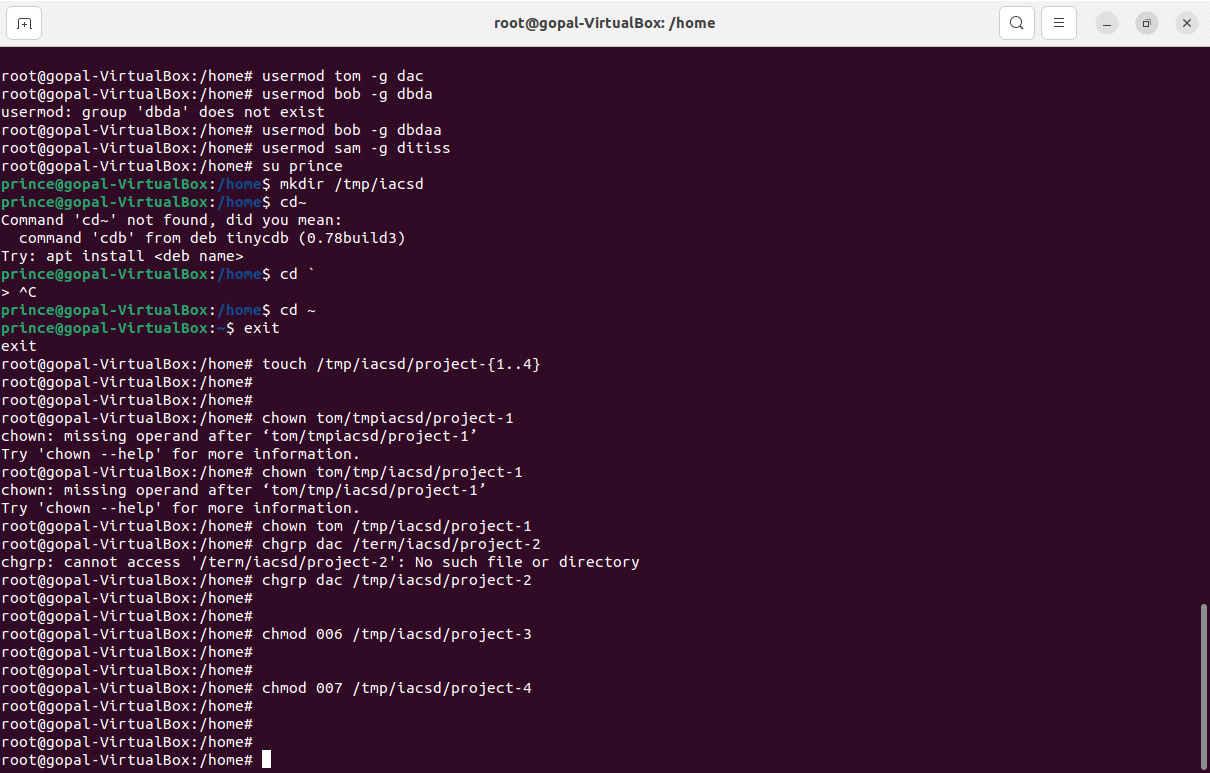
access

Project-4 – dbda group should have rwx permissions.

**chown tom /tmp/iacsd/project-1**

**chgrp dac /tem/iacsd/project-2**

**chmod 006 /tmp/iacsd/project-3**

**chmod 007 /tmp/iacsd/project-4**

**ASS-4**

1) Write a shell script tp print

● your are logged in as which user

● in which directory you are

● and in which terminal you are working

● total number of files and directories in current directory

ass#!/bin/bash

# Get the current username

current\_user=$(whoami)

# Get the current directory

current\_directory=$(pwd)

# Get the terminal name

terminal\_name=$(tty)

# Count the number of files and directories in the current directory

file\_count=$(ls -l | grep -v total | wc -l)

directory\_count=$(ls -l | grep "^d" | wc -l)

# Print the information

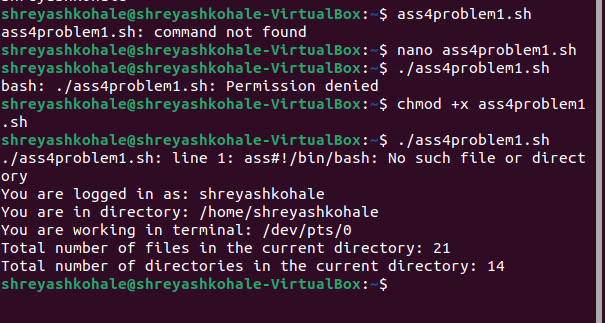
echo "You are logged in as: $current\_user"

echo "You are in directory: $current\_directory"

echo "You are working in terminal: $terminal\_name"

echo "Total number of files in the current directory: $file\_count"

echo "Total number of directories in the current directory: $directory\_count"



2).Write a shell script to create a menu driven program for adding, deletion or

finding a record in a database. Database should have the field like rollno, name,

semester and marks of three subjects. Last option of the menu should be to exit the

menu.

#!/bin/bash

# Initialize an associative array to store records

declare -A database

# Menu loop

while true; do

echo "Database Menu:"

echo "1. Add Record"

echo "2. Delete Record"

echo "3. Find Record"

echo "4. Exit"

read choice

case $choice in

1)

echo "Enter Roll No:"

read rollno

echo "Enter Name:"

read name

echo "Enter Semester:"

read semester

echo "Enter Marks for 3 Subjects (space-separated):"

read -a marks

# Store the record in the database

database["$rollno"]="$name $semester ${marks[\*]}"

echo "Record added.";;

2)

echo "Enter Roll No to delete:"

read rollno

if [ -n "${database[$rollno]}" ]; then

unset database["$rollno"]

echo "Record deleted."

else

echo "Record not found."

fi;;

3)

echo "Enter Roll No to find:"

read rollno

if [ -n "${database[$rollno]}" ]; then

echo "Record: ${database[$rollno]}"

else

echo "Record not found."

fi;;

4)

echo "Exiting the menu."

exit;;

\*)

echo "Invalid option. Please choose 1, 2, 3, or 4.";;

esac

done



3) Write a Linux shell script to accept 10 number and tell how many are +tive, -

tive and zero.

#!/bin/bash

positive=0

negative=0

zero=0

for ((i=1; i<=10; i++))

do

echo "Enter number $i:"

read num

if [ $num -gt 0 ]; then

positive=$((positive + 1))

elif [ $num -lt 0 ]; then

negative=$((negative + 1))

else

zero=$((zero + 1))

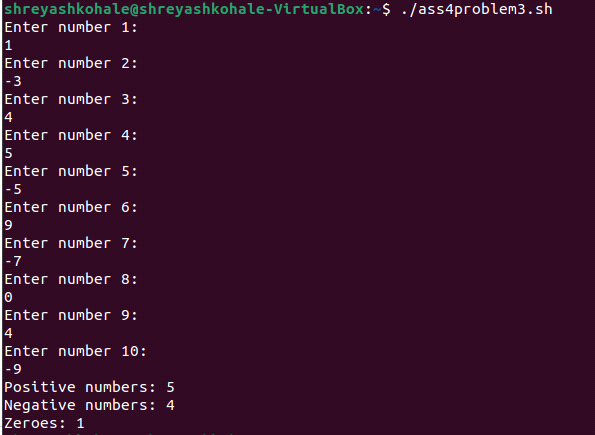
fi

done

echo "Positive numbers: $positive"

echo "Negative numbers: $negative"

echo "Zeroes: $zero"



4) Write a shell script to accept five number and display max and min value.

#!/bin/bash

max=0

min=0

for ((i=1; i<=5; i++))

do

echo "Enter number $i:"

read num

if [ $i -eq 1 ]; then

max=$num

min=$num

else

if [ $num -gt $max ]; then

max=$num

fi

if [ $num -lt $min ]; then

min=$num

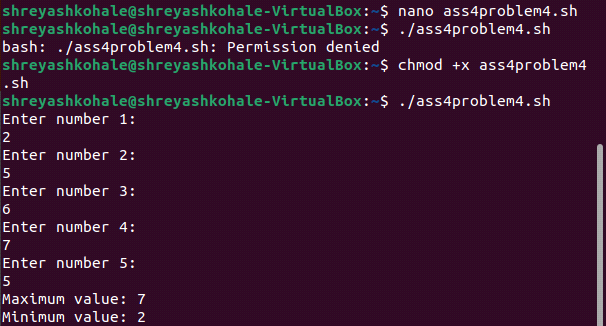
fi

fi

done

echo "Maximum value: $max"

echo "Minimum value: $min"



5) Write a script to find out String is palindrome or not.

#!/bin/bash

echo "Enter a string:"

read input\_string

# Remove spaces and convert to lowercase for case-insensitive comparison

clean\_string=$(echo "$input\_string" | tr -d '[:space:]' | tr '[:upper:]' '[:lower:]')

# Reverse the string

reversed\_string=$(echo "$clean\_string" | rev)

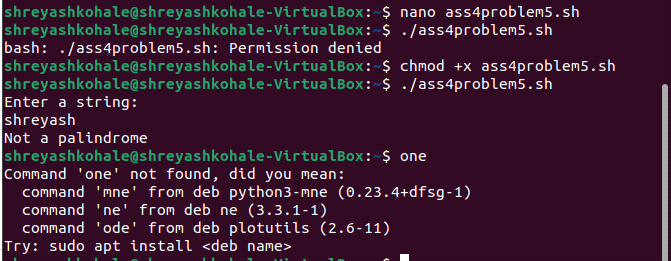
if [ "$clean\_string" == "$reversed\_string" ]; then

echo "Palindrome"

else

echo "Not a palindrome"

fi



6) Write a shell script to print given number’s sum of all digits (eg. If number is

123, then it’s sum of all digits will be 1+2+3=6)

#!/bin/bash

echo "Enter a number:"

read num

sum=0

while [ $num -gt 0 ]; do

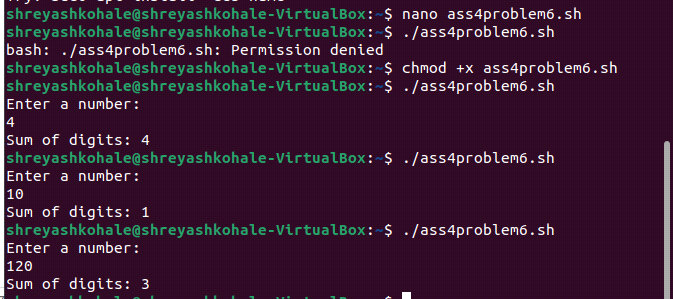
digit=$((num % 10))

sum=$((sum + digit))

num=$((num / 10))

done

echo "Sum of digits: $sum"



7) Create a script to

Create user , Delete user , Create group , delete Group using case

#!/bin/bash

# Menu loop

while true; do

echo "User and Group Management Menu:"

echo "1. Create User"

echo "2. Delete User"

echo "3. Create Group"

echo "4. Delete Group"

echo "5. Exit"

read choice

case $choice in

1)

echo "Enter username to create:"

read username

useradd "$username"

echo "User $username created.";;

2)

echo "Enter username to delete:"

read username

userdel -r "$username" # -r removes the user's home directory

echo "User $username deleted.";;

3)

echo "Enter group name to create:"

read groupname

groupadd "$groupname"

echo "Group $groupname created.";;

4)

echo "Enter group name to delete:"

read groupname

groupdel "$groupname"

echo "Group $groupname deleted.";;

5)

echo "Exiting the menu."

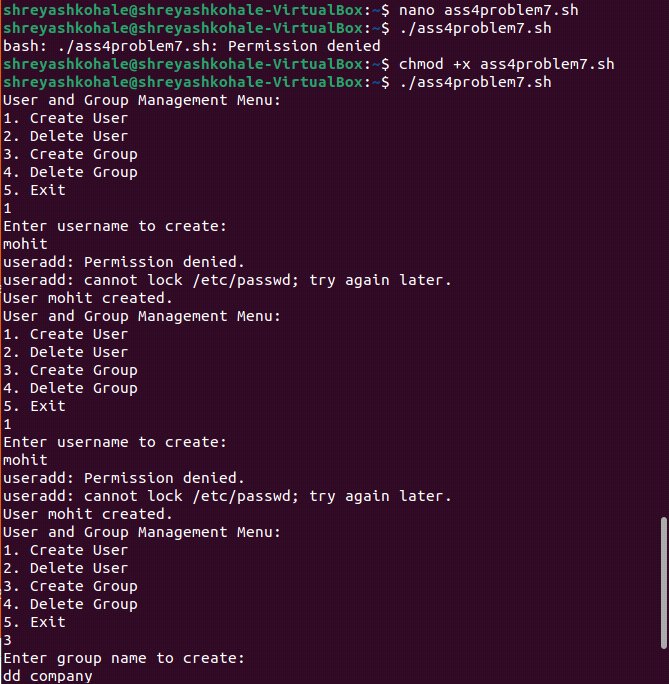
exit;;

\*)

echo "Invalid option. Please choose 1, 2, 3, 4, or 5.";;

esac

done



**ASSIGNMENT1 TXT**

sudo addgroup one

sudo adduser apple --gid 1001

sudo adduser banana --gid 1001

sudo adduser cat --gid 1001

sudo adduser dog --gid 1001

sudo adduser elephant --gid 1001

sudo addgroup two

sudo adduser fish --gid 1002

sudo adduser gun --gid 1002

sudo adduser horse --gid 1002

sudo adduser icecream --gid 1002

sudo addgroup three

sudo adduser jelly --gid 1003

sudo adduser kitkat --gid 1003

sudo adduser lolipop --gid 1003

sudo adduser marshmallow --gid 1003

sudo addgroup four

sudo adduser new --gid 1004

sudo adduser oppo --gid 1004

sudo adduser vivo --gid 1004

sudo adduser china --gid 1004

/home -> mkdir EVERYONE

cd /home

mkdir EVERYONE

sudo chmod 777 EVERYONE

Create a file with every user (whoami >> username.txt)

jelly,kitkat, lolipop, marshmallow -> add these users to sudo group

sudo usermod oppo -g one

sudo usermod vivo -g two

sudo usermod jelly -G sudo

sudo usermod kitkat -G sudo

sudo usermod lollipop -G sudo

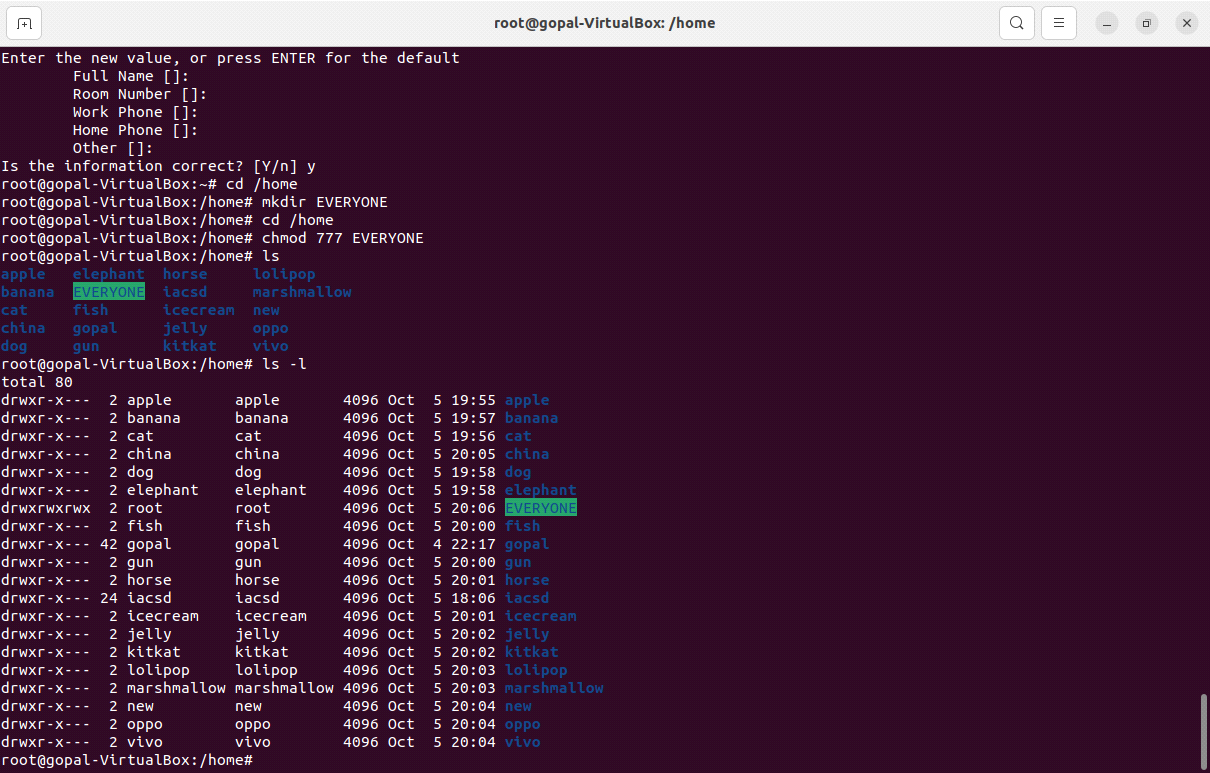
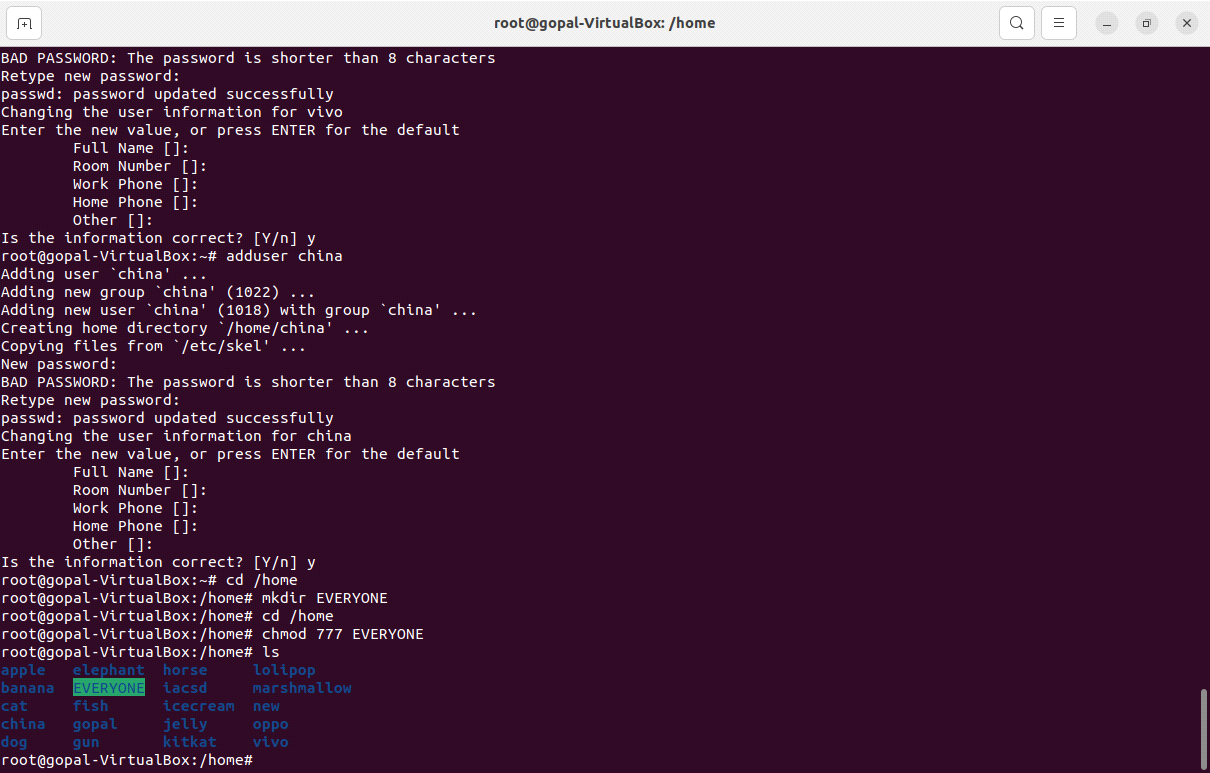
sudo usermod marshmallow -G sudo

cat /etc/group

fish,gun -> add these users to one group as well (secondary group)

sudo usermod fish -G one

sudo usermod gun -G one



**Exercise**

**#first**

#program to find maximum number between two numbers

#! /bin/bash

read -p "Enter first number" num1

read -p "Enter second number" num2

if [ $num1 -gt $num2 ]

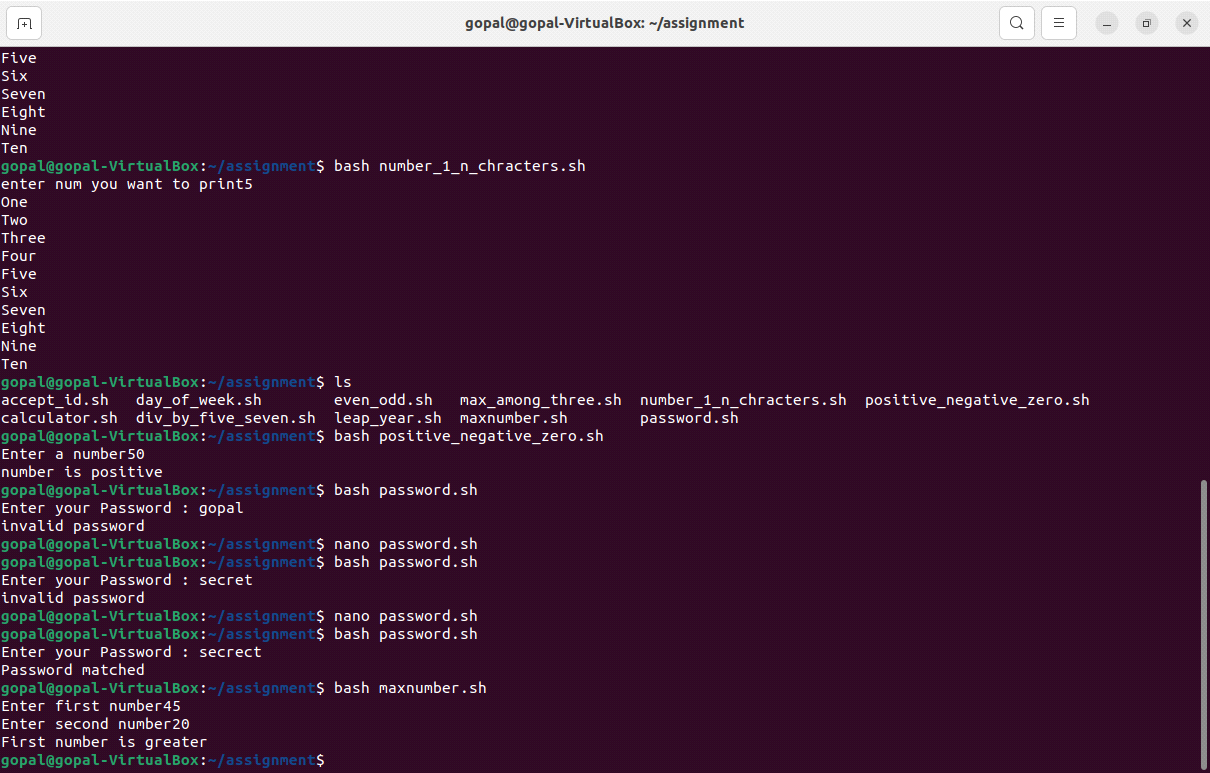
then

echo "First number is greater"

else

echo "Second number is greater"

fi



**#second**

#program to find maximum number among threee numbers

#! /bin/bash

read -p "Enter n1" n1

read -p "Enter n2" n2

read -p "Enter n3" n3

if [ $n1 -gt $n2 ]

then

if [ $n1 -gt $n3 ]

then

echo "n1 is greater"

else

echo "n3 is greater"

fi

else

if [ $n2 -gt $n3 ]

then

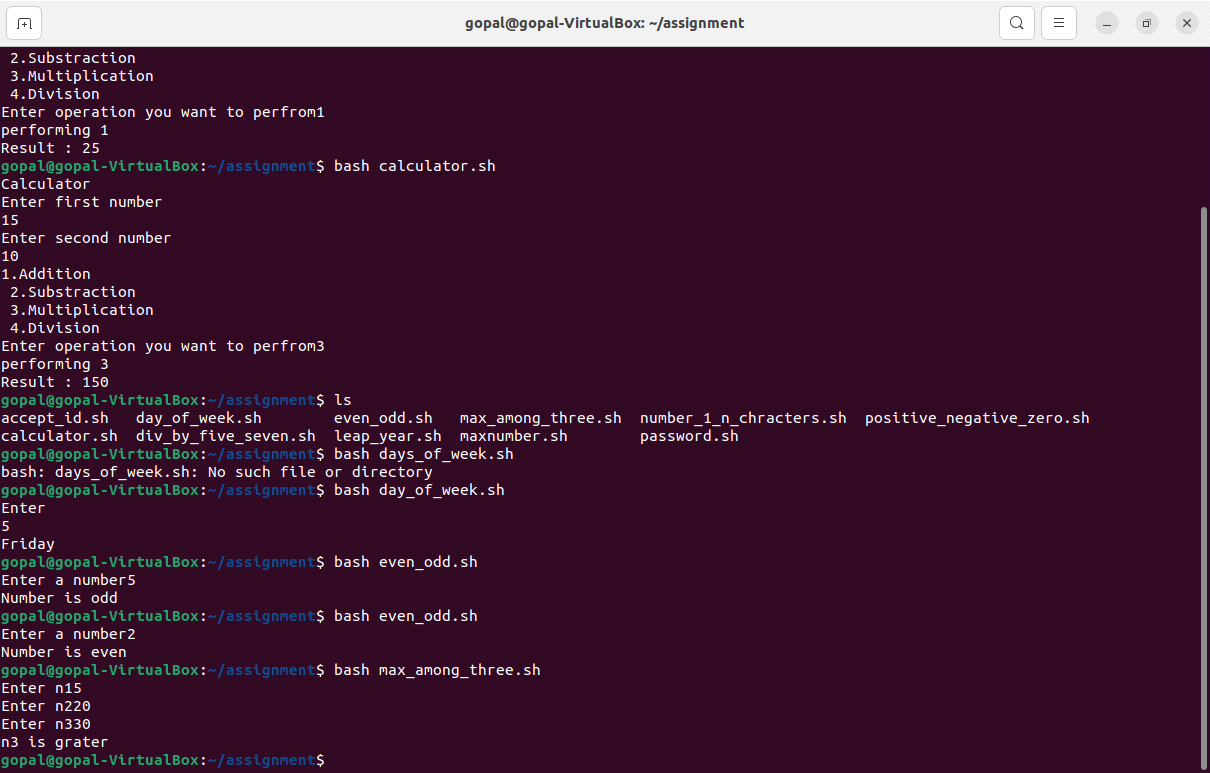
echo "n2 is greater"

else

echo "n3 is grater"

fi

fi



**#third**

#! /bin/bash

read -p "Enter a number" number

if [ $number -gt 0 ]

then

echo "number is positive"

else

if [ $number -eq 0 ]

then

echo "number is equal to zero"

else

echo "number is negative"

fi

fi

**#fourth**

#! /bin/bash

read -p "Enter a number" number

if [ $number -gt 0 ]

then

echo "number is positive"

else

if [ $number -eq 0 ]

then

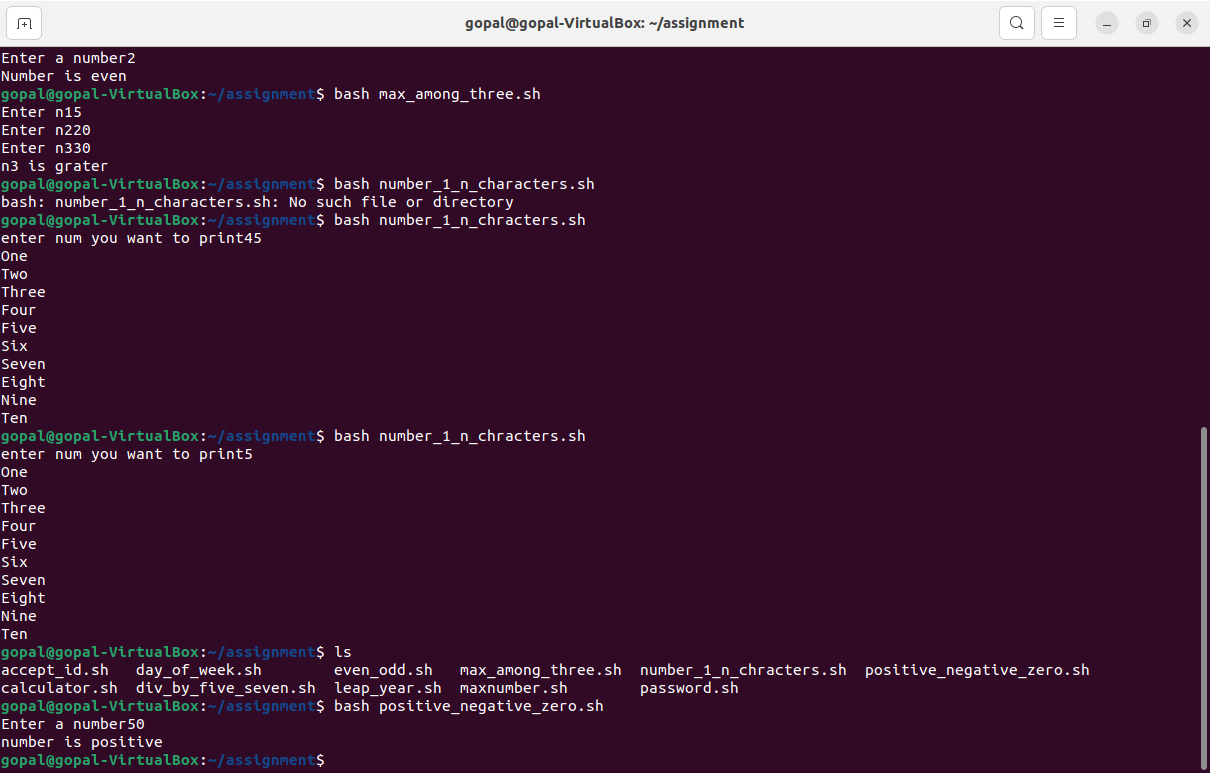
echo "number is equal to zero"

else

echo "number is negative"

fi

fi



**#fifth**

#! /bin/bash

read -p "Enter a number" number

if [ `expr $number % 2` == 0 ]

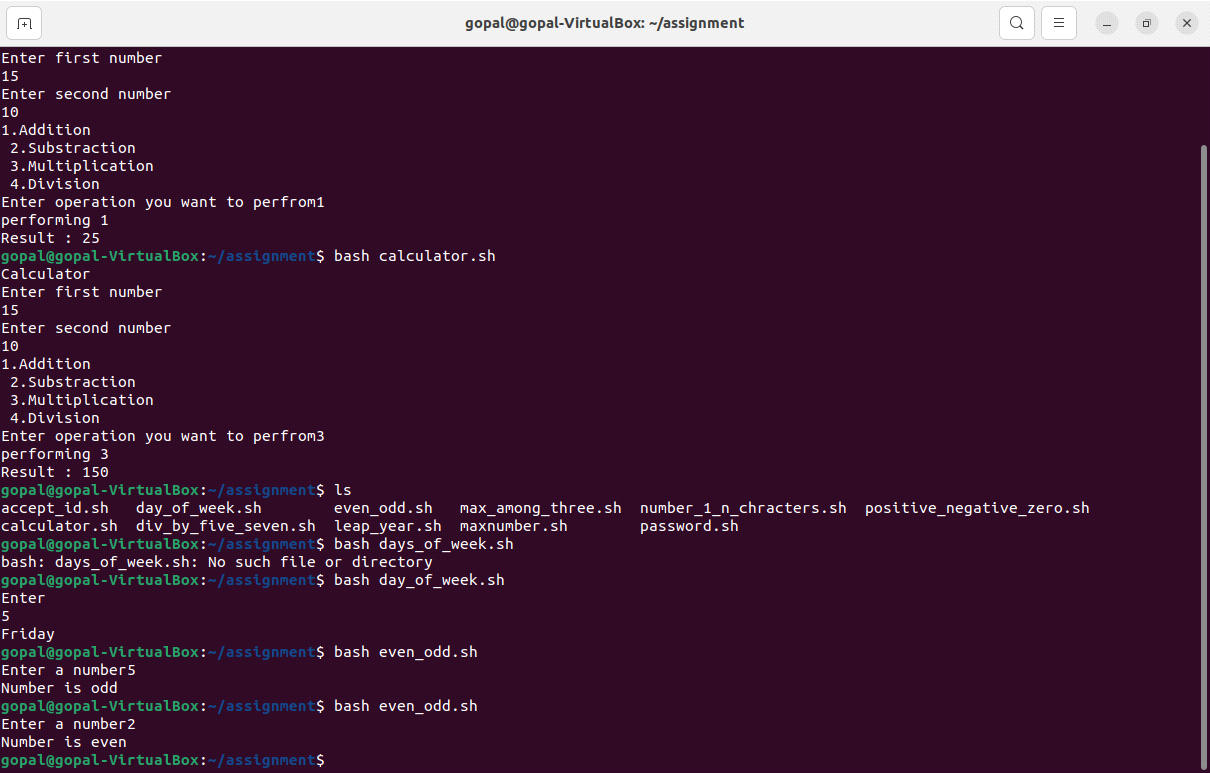
then

echo "Number is even"

else

echo "Number is odd"

fi



**#sixth**

#! /bin/bash

read -p "Enter year" year

if [ `expr $year % 400` == 0 ] || [ `expr $year % 100` != 0 ] && [ `expr $year % 4` == 0 ]

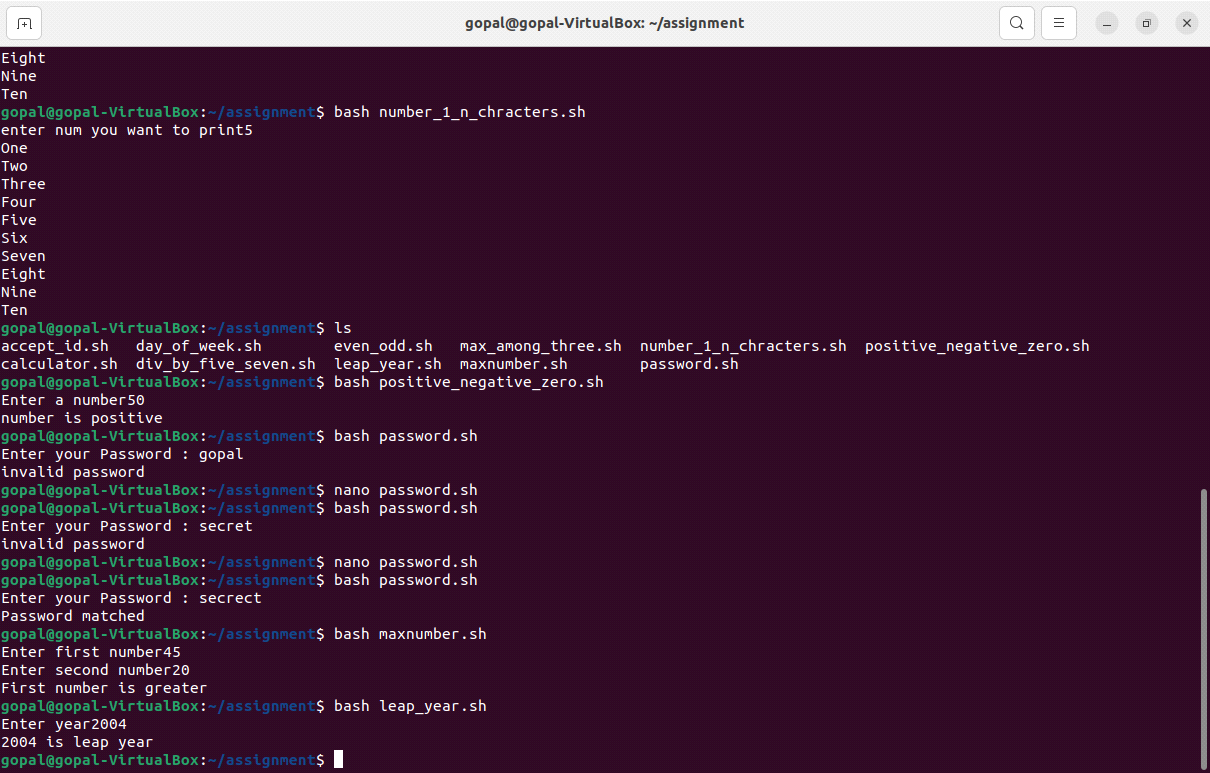
then

echo $year "is leap year"

else

echo $year "is not leap year"

fi



**#seventh**

#! /bin/bash

read -p "enter num you want to print" num

for num in {1..10}; do

case $num in

1)

echo "One"

;;

2)

echo "Two"

;;

3)

echo "Three"

;;

4)

echo "Four"

;;

5)

echo "Five"

;;

6)

echo "Six"

;;

7)

echo "Seven"

;;

8)

echo "Eight"

;;

9)

echo "Nine"

;;

10)

echo "Ten"

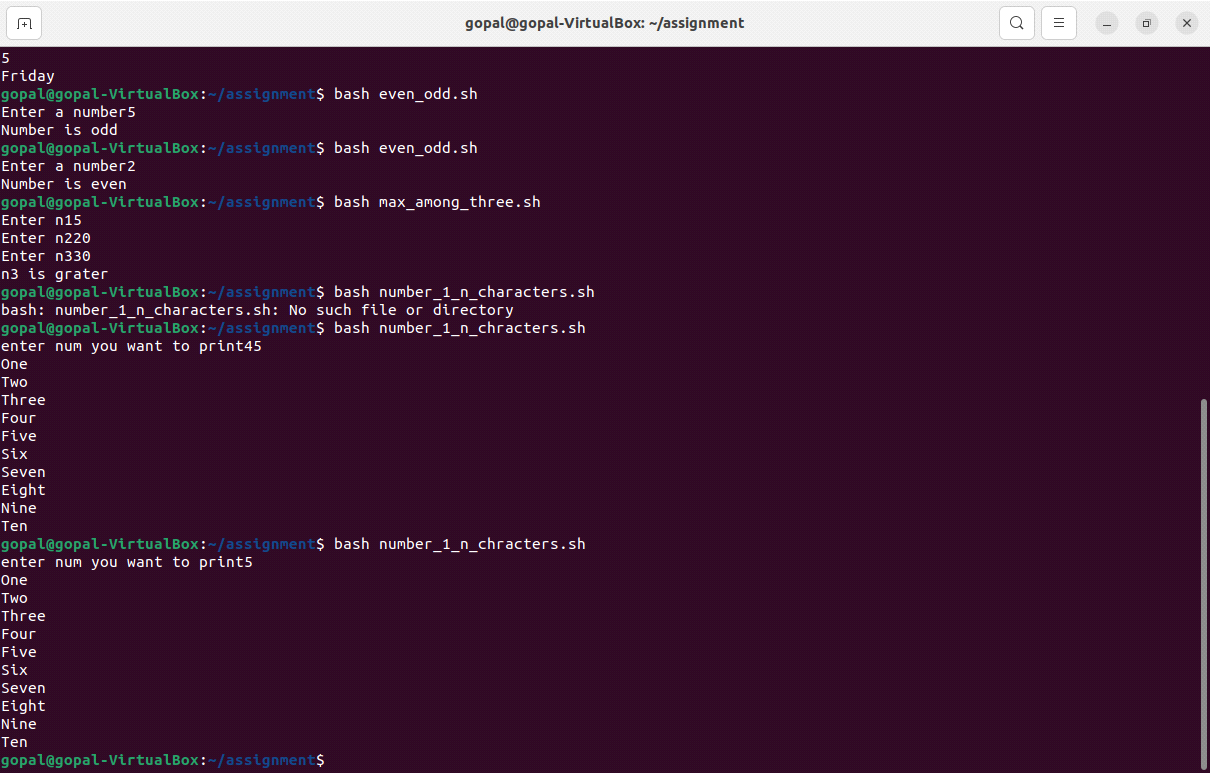
;;

\*)

echo "number out of range"

esac

done



**#eighth**

#! /bin/bash

read -p "Enter your deptID" id

case $id in

100)

echo "HR department"

;;

101)

echo "Finance Department"

;;

102)

echo "Technical Department"

;;

103)

echo "Marketing Department"

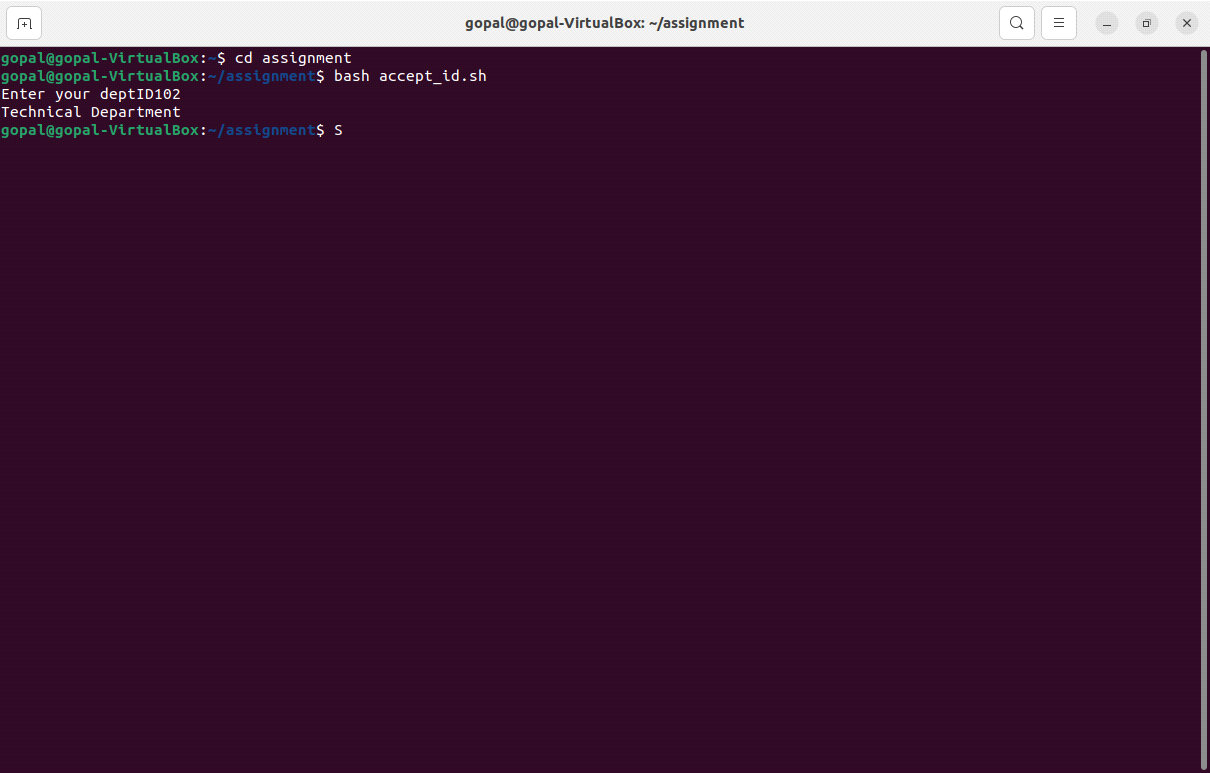
;;

\*)

echo "Department not found"

;;

esac



**#ninth**

#! /bin/bash

password="secrect"

read -p "Enter your Password : " pass

case "$pass" in

"$password")

echo "Password matched"

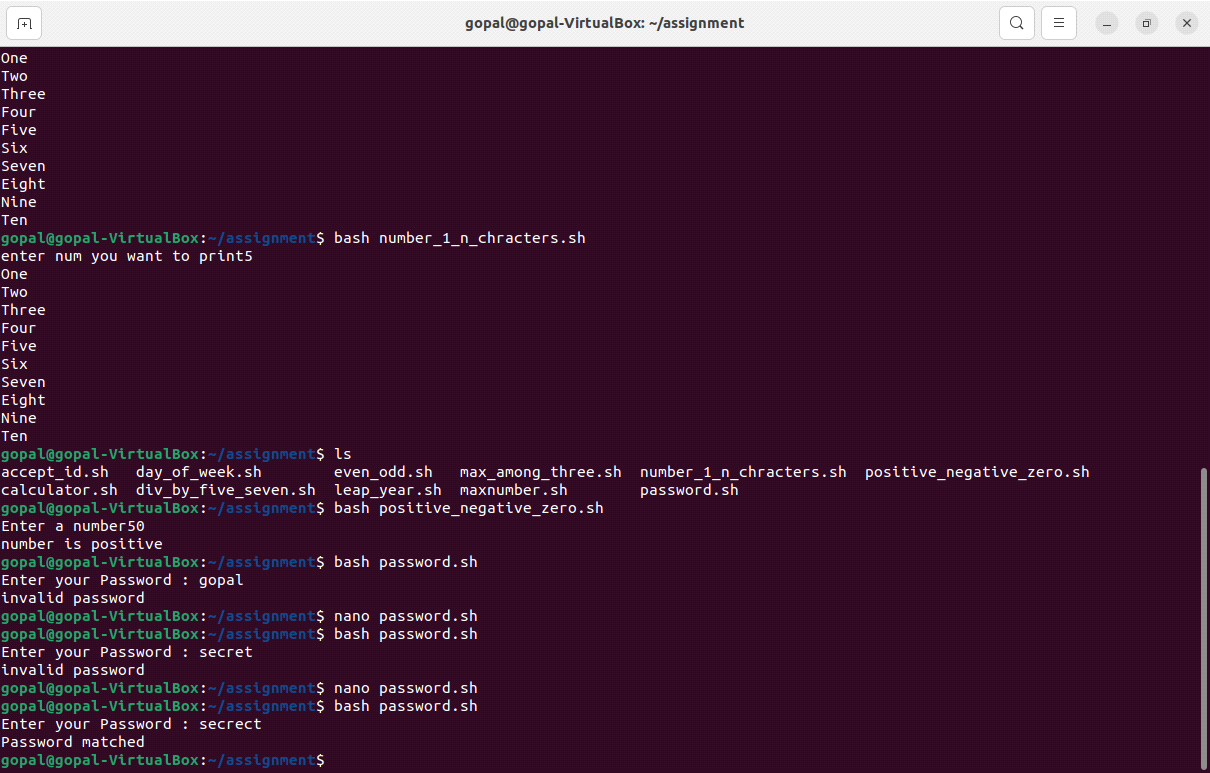
;;

\*)

echo "invalid password"

;;

esac



**#tenth**

#! /bin/bash

#read-p#weekdays of week" $week

echo "Enter"

read num

case $num in

1)

echo "Monday"

;;

2)

echo "Tuesday"

;;

3)

echo "Wednesday"

;;

4)

echo "Thursday"

;;

5)

echo "Friday"

;;

6)

echo "Saturday"

;;

7)

echo "Sunday"

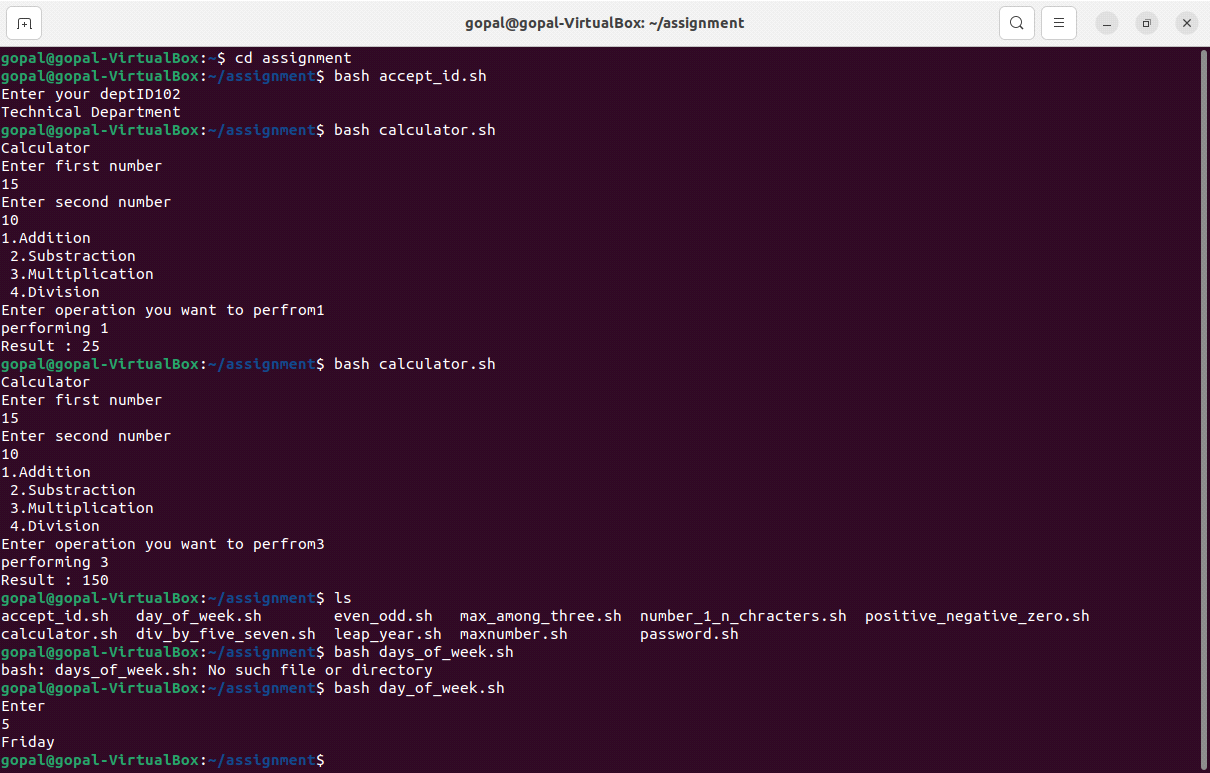
;;

\*)

echo "Invalid input"

;;

esac



**#eleventh**

#! /bin/bash

echo "Calculator"

echo "Enter first number"

read num1

echo "Enter second number"

read num2

echo -e "1.Addition \n 2.Substraction \n 3.Multiplication \n 4.Division "

read -p "Enter operation you want to perfrom" operation

echo "performing $operation"

case $operation in

1)

result=$((num1+num2))

add="Addition"

;;

2)

result=$((num1-num2))

sub="Substraction"

;;

3)

result=$((num1\*num2))

mul="Multiplication"

;;

4)

result=$((num1/num2))

div="Division"

;;

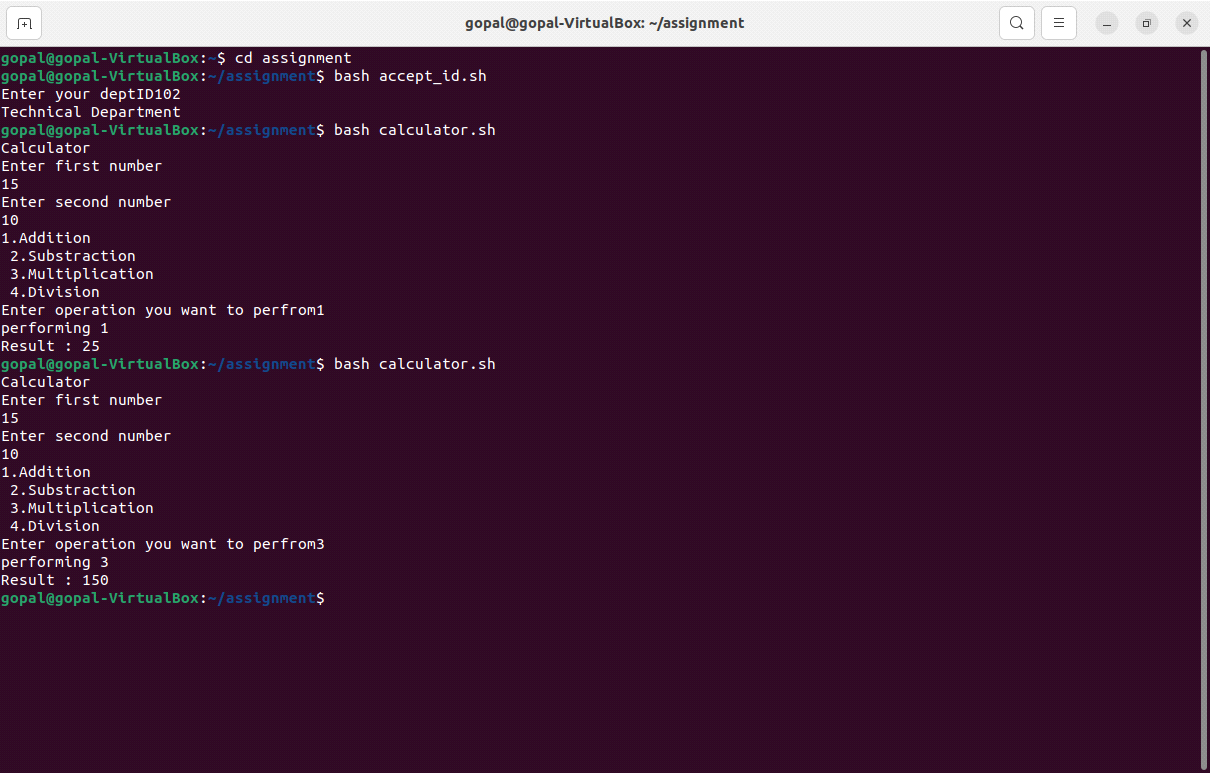
\*)

echo "Invalid Operation"

;;

esac

echo "Result : $result"



**LOOPS ASSIGNMENT**

﻿1. Shell Script to display the first 10 natural numbers.

Expected Output :

1 2 3 4 5 6 7 8 9 10

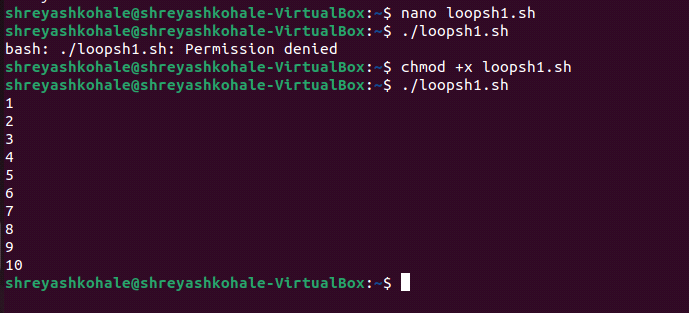
#!/bin/bash

for num in {1..10}

do

echo "$num "

done



2. Shell Script to compute the sum of the first 10 natural numbers.

Expected Output :

The first 10 natural number is :

1 2 3 4 5 6 7 8 9 10

The Sum is : 55

#!/bin/bash

echo "first 10 natural number is :"

sum=0

for ((i=1; i<=10; i++))

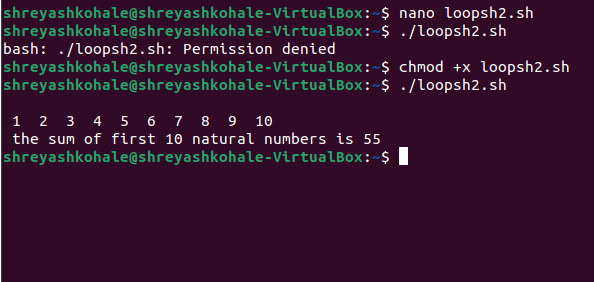
do

echo -n " $i "

sum=$((sum + i))

done

echo -e "\n the sum of first 10 natural numbers is $sum"



3. Shell Script to display n terms of natural numbers and their sum.

Test Data : 7

Expected Output :

The first 7 natural number is :

1 2 3 4 5 6 7

The Sum of Natural Number upto 7 terms : 28

#!/bin/bash

echo "Enter the value of n:"

read n

sum=0

number=""

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

do

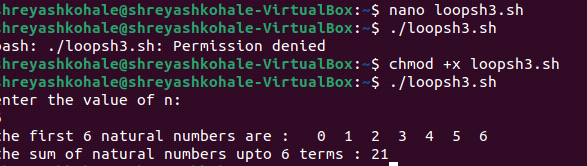
numbers="$numbers $i"

sum=$((sum + i))

done

echo "The first $n natural numbers are: $numbers"

echo "The sum of natural numbers upto $n terms: $sum"



4. Shell Script to read 10 numbers from the keyboard and find their sum and average.

Test Data :

Input the 10 numbers :

Number-1 :2

...

Number-10 :2

Expected Output :

The sum of 10 no is : 55

The Average is : 5.500000

#!/bin/bash

sum=0

count=10

echo "input 10 numbers"

for ((i=1;i<=count;i++))

do

echo -n "Number-$i : "

read num

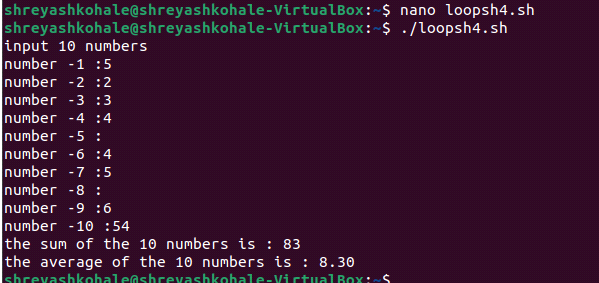
sum=$((sum + num))

done

average=$(echo "scale=2; $sum/$count" | bc)

echo "The sum of the 10 numbers is : $sum"

echo "The average of the 10 numbers is : $average"



5. Shell Script to display the cube of the number up to an integer.

Test Data :

Input number of terms : 5

Expected Output :

Number is : 1 and cube of the 1 is :1

Number is : 2 and cube of the 2 is :8

Number is : 3 and cube of the 3 is :27

Number is : 4 and cube of the 4 is :64

Number is : 5 and cube of the 5 is :125

#!/bin/bash

echo "Input number of terms: "

read n

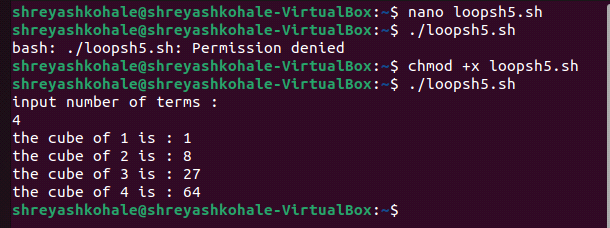
for((i=1;i<=n;i++))

do

cube=$((i\*i\*i))

echo "The cube of $i is : $cube"

done



6. Shell Script to display the multiplication table for a given integer.

Test Data :

Input the number (Table to be calculated) : 15

Expected Output :

15 X 1 = 15

...

#!/bin/bash

echo "Input the number to calculate table : "

read number

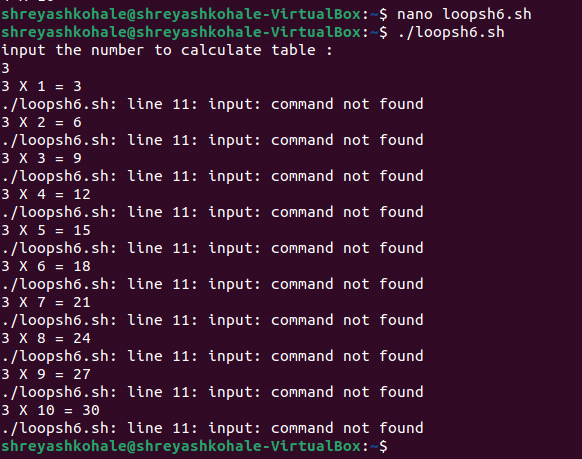
for((i=1;i<=10;i++))

do

product=$((number\*i))

echo "$number X $i = $product"

done

...

15 X 10 = 150

7. Shell Script to display the multiplier table vertically from 1 to n.

Test Data :

Input upto the table number starting from 1 : 8

Expected Output :

Multiplication table from 1 to 8

1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8

...

1x10 = 10, 2x10 = 20, 3x10 = 30, 4x10 = 40, 5x10 = 50, 6x10 = 60, 7x10 = 70, 8x10 = 80

#!/bin/bash

echo "Input upto the table number starting from 1 :"

read number

for((i=1;i<=number;i++))

do

for((j=1;j<=10;j++))

do

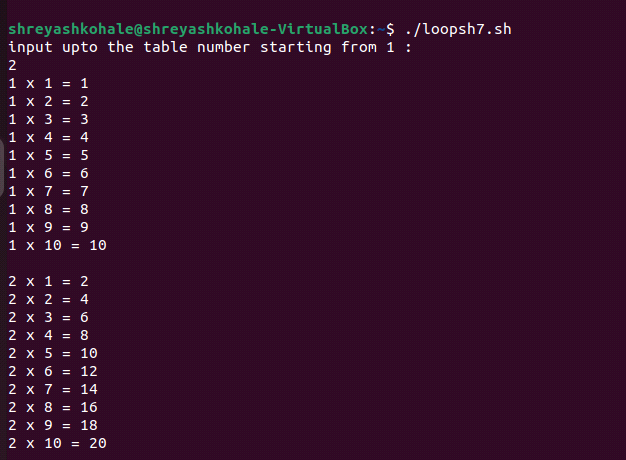
product=$((i\*j))

echo " $i x $j = $product"

done

echo

done



8. Shell Script to display the n terms of odd natural numbers and their sum.

Test Data

Input number of terms : 10

Expected Output :

The odd numbers are :1 3 5 7 9 11 13 15 17 19

The Sum of odd Natural Number upto 10 terms : 100

#!/bin/bash

echo "Enter the number of terms :"

read n

sum=0

count=0

odd\_number=""

current\_number=1

while [ $count -lt $n ]

do

if [ $((current\_number % 2)) -ne 0 ]; then

odd\_number="odd\_numbers $current\_number"

sum=$((sum + current\_number))

count=$((count + 1))

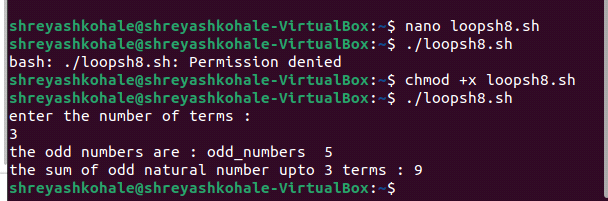
fi

current\_number=$((current\_number + 1))

done

echo "The odd numbers are : $odd\_number"

echo "The sum of odd natural number upto $n terms : $sum"



9. Shell Script to display a pattern like a right angle triangle using an asterisk.

The pattern like :

\*

\*\*

\*\*\*

\*\*\*\*

#!/bin/bash

echo "Enter the number of rows for pattern :"

read n

for ((i=1;i<=n;i++))

do

for ((j=1;j<=i;j++))

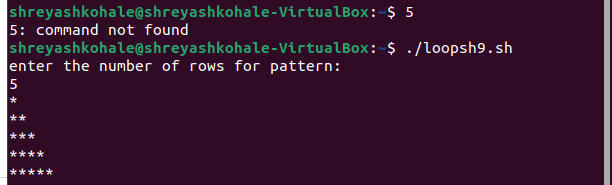
do

echo -n "\*"

done

echo

done



10. Shell Script to display a pattern like a right angle triangle with a number.

The pattern like :

1

12

123

1234

#!/bin/bash

echo "Enter the no of rows for pattern :"

read n

echo

for ((i=1;i<=n;i++))

do

for ((j=1;j<=i;j++))

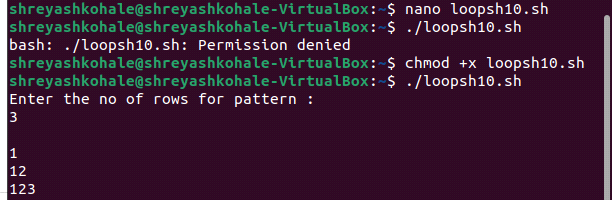
do

echo -n "$j"

done

echo

done



11. Shell Script to make such a pattern like a right angle triangle with a number which will repeat a number in a row.

The pattern like :

1

22

333

4444

#!/bin/bash

echo "Enter the number of rows you want for the pattern: "

read n

echo

for ((i=1;i<=n;i++))

do

for ((j=1;j<=i;j++))

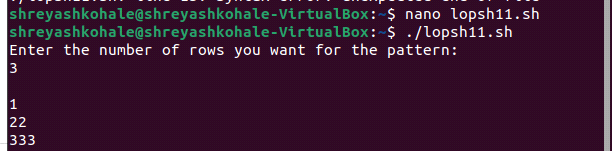
do

echo -n "$i"

done

echo

done



12. Shell Script to make such a pattern like a right angle triangle with the number increased by 1.

The pattern like :

1

2 3

4 5 6

7 8 9 10

#!/bin/bash

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

read n

echo

currentnumber=1

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

do

for ((j=0;j<=i;j++))

do

echo -n "$currentnumber "

currentnumber=$((currentnumber+1))

done

echo

done

