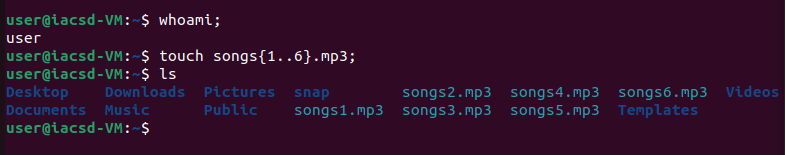
## **ASSIGNMENT: 1**

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

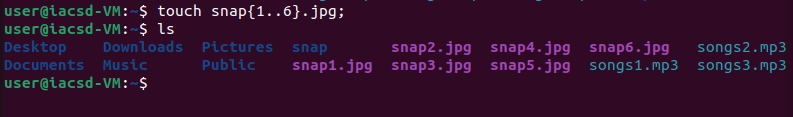
**● Create 6 files with names of the form songsX.mp3.**

touch songs{1..6}.mp3



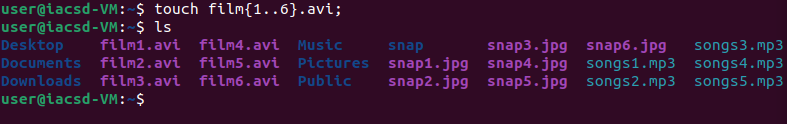
**● Create 6 files with names of the form snapX.jpg.**

touch snap{1..6}.jpg



**● Create 6 files with names of the form filmX.avi.**

touch film{1..6}.avi

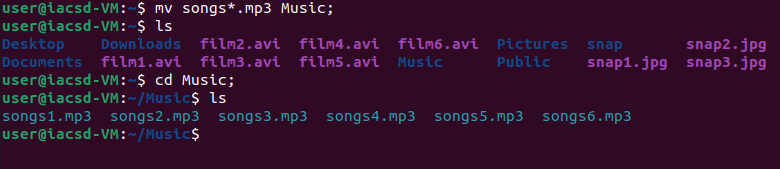


**In each set, replace X with the numbers 1 through 6.**

**2. From your home directory,**

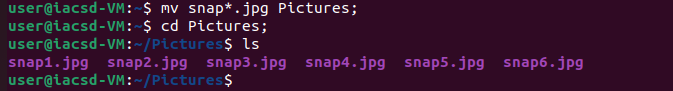
**● Move songs file into your Music subdirectory.**

mv songs\*.mp3 Music



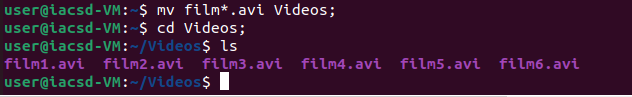
**● Move snap file into your Pictures subdirectory.**

mv snap\*.jpg Picture



**● Move your movie files into Videos subdirectory**

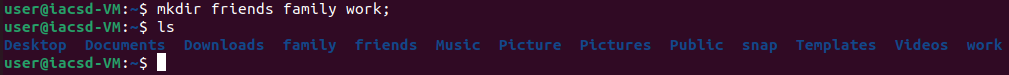
mv film\*.avi Videos



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

***To Create friends, family, and work subdirectories***

mkdirfriends 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.**

**Copy files with numbers 1 and 2 to the friends folder**

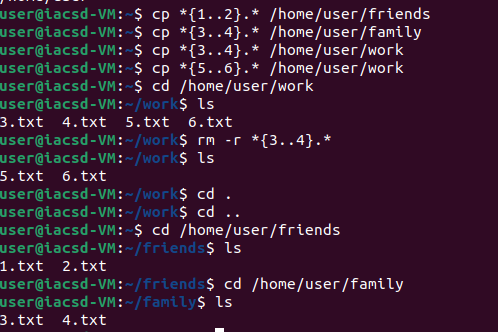
cp \*{1,2}.\* friends mt

**Copy files with numbers 3 and 4 to the family folder**

cp \*{3,4}.\* family

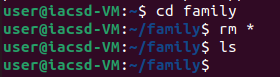
**Copy files with numbers 5 and 6 to the work folder**

cp \*{5,6}.\* work

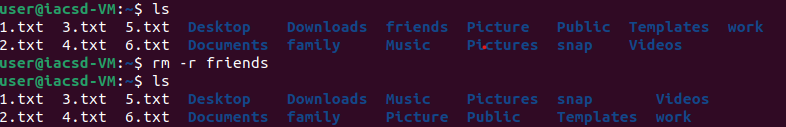


# **ASSIGNMENT: 2**

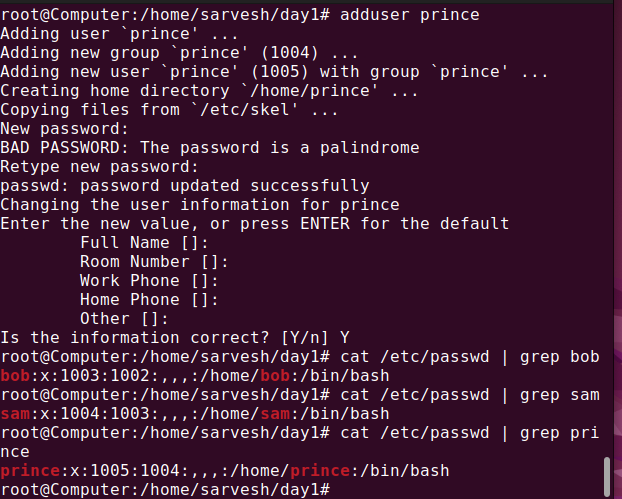
**6. Delete all files in family subdirectory.**



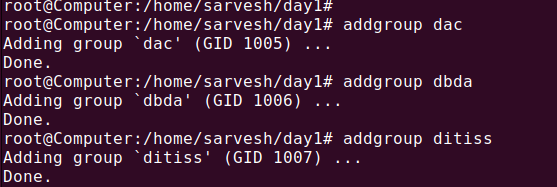
**7. Delete friends subdirectory**



**8. Create user tom , bob , sam , prince**



**9. Create Group dac ,dbda ,ditiss**



**10. add user**

**Tom in dac**

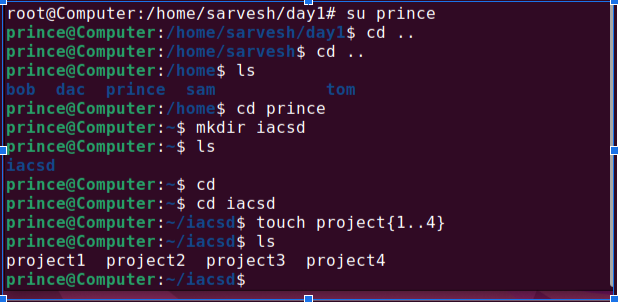
**Bob in dbda**

**Sam in ditiss**

A screenshot of a computer program

Description automatically generated

**11. login as prince and create iacsddirectory in /tmp and create 4 files in iacsd with name project-1 project-2 upto 4**



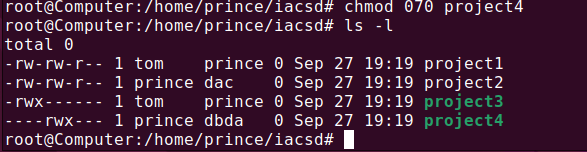
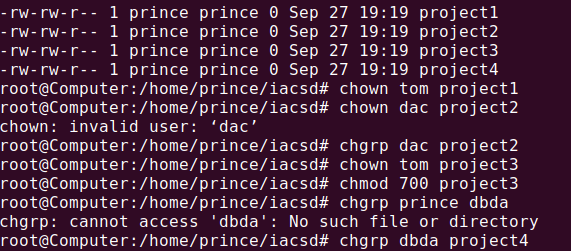
**12. assign permissions to project files as below**

**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.**

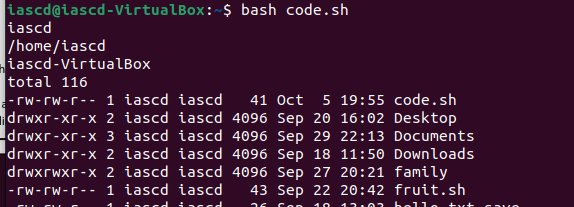


# **ASSIGNMENT: 3**

**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**



****

**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

n=1

echo "select 1) add 2) delete 3) find 4)exit"

while [ $n -ne 4 ]

do

read -p "enter choise: " n

case $n in

1)

read -p "enter roll: " rollno

read -p "enter name: " name

read -p "enter sem : " sem

echo -n "$rolino $name $sem " >> data.txt

for((i=1;i<4;i++))

do

read -p "enter marks $i :" m

echo -n "$m">>data.txt

done

echo "" >>data.txt

;;

2)

read -p "enter name : " name

sed '/name/d' data.txt

cat data.txt

;;

3)

read -p "enter roll : " r

cat data.txt|grep $r

;;

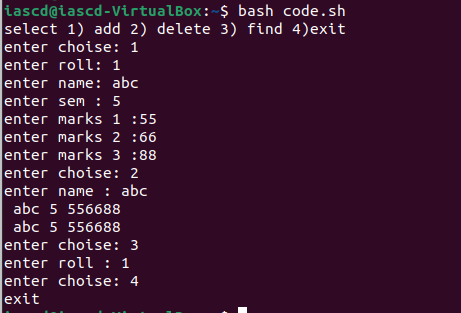
4)

echo "exit"

;;

esac

done



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

#!/bin/bash

read -p "Enter size of array : " size

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

do

read -p "enter element : " n

if [ $n -gt0 ]

then

echo $n >> pos.txt

elif [ $n -lt0 ]

then

echo $n >> neg.txt

else

echo $n >> zero.txt

fi

done

wc -w pos.txt

echo "positive no : "

cat pos.txt

rm pos.txt

echo "-------"

wc -w zero.txt

echo "zero no : "

cat zero.txt

rm zero.txt

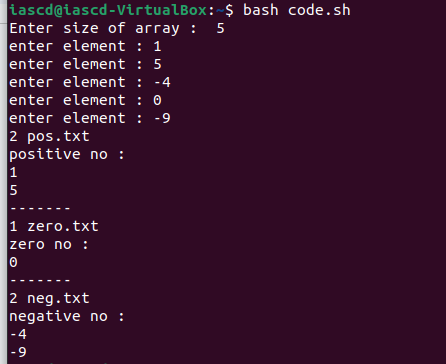
echo "-------"

wc -w neg.txt

echo "negative no : "

cat neg.txt

rm neg.txt



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

#!/bin/bash

for((i=0;i<5;i++))

do

read -p "enter no : " n

echo $n >> data.txt

done

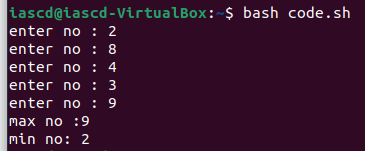
echo -n "max no :"

sort -r data.txt | head -n 1

echo -n "min no: "

sort data.txt |head -n 1

rm data.txt



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

**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

read -p "enter no : " no

while [ $no -ne 0 ]

do

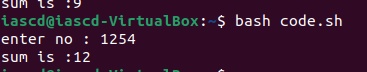
sum=$((sum+no%10))

r=$((no/10))

no=$r

done

echo "sum is :$sum"



**7) Create a script to**

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

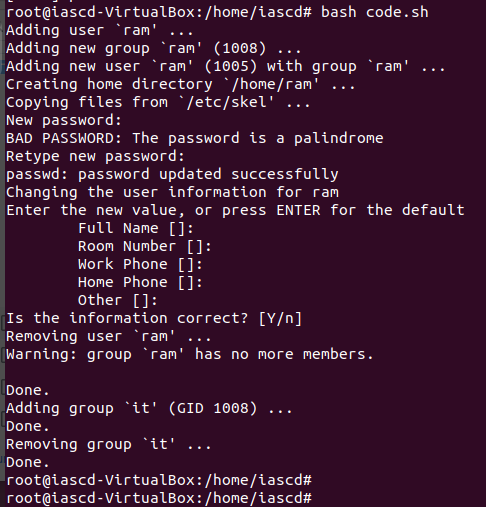
#!/bin/bash

adduser ram

deluser ram

addgroup it

delgroup it



# **ASSIGNMENT:4**

# **ASSIGNMENT: 4**

pwd

mkdir EVERYONE

pwd

cd EVERYONE

pwd

adduser one

sudoadduser one

groups

cat /etc

cat/ etc/group

cat /etc/groups

etc/group

etc

cd cat/etc/group

ls -l

ls-lt

pwd

user

users

cat /etc/passwd

clear

pwd

whoami

su one

sudo --gid 1001

clear

pwd

cd

pwd

cd home

pwd

su one

whmio

whoami

su one

pwd

cd EVERYONE

pwd

ls

chmod EVERYONE

chmod 777 EVERYONE

usermod

clear

one

exit

ls

pwd

cd /home

ls

cat /etc/group

cat /etc/passwd

cd

mkdir EVERYONE

cd EVERYONE

chmod 777 EVERYONE

ls -l

sudochmod 777 EVERYONE

cd ../

ls

chmod EVERYONE

chmod 777 EVERYONE

chmod

ls

ls -l EVERYONE

cat /etc/passwd

userdel one

sudouserdel one

clear

cd

cd EVERYONE

ls -l

passwd

userfiles

sudo everyone

sudEVERYONE

ls EVERYINE

ls EVERYONE

sudo root

ls -s

ls -

cd

ls

ls -l

cd EVERYONE/

addgroup ONE

sudoaddgroup ONE

add group ONE

cd../

cd ..

addgroup ONE

sudoaddgroup ONE

sudoaddgroup one

cd one

ls

cat /etc/group

add user --gid1001 apple

cat /etc/group

adduser apple

sudoadduser apple

cat/etc/group

cat /etc/group

usermod apple -G

usermod apple -G one

sudousermod apple -G one

cat /etc/group

adduser banana

sudoadduser banana

sudousermod banana -G one

cat /etc/groups

cat /etc/group

sudoadduser cat

y

sudousermod cat -

sudousermod cat -G one

cat /etc/group

usermod cat --gid one

sudousermod cat --gid one

cat /etc/group

cat/ passwd

cat /etc/passwd

cat /etc/group

sudoadduser dog

sudousermod dog -G one

sudoadduser elephant

sudousermod elephant -G one

cat /etc/group

sudoaddgroup two

sudoadduser fish

sudousermod fish -

sudousermod fish -G two

sudoadduser gun

sudousermod gun -G two

sudoadduser horse

sudousermod horse -G two

sudoaddusericecream

sudousermodicecream -G two

cat /etc/group

su two

chgrp two

ls -l

sudochgrp

sudochgrp one

sudoaddgroup three

sudoadduser jelly

sudousermod jelly -G three

sudoadduserkitkat

sudousermodkitkat -G three

sudoadduser lollipop

cat /etc/group

sudousermod lollipop -G three

sudoaddgroup four

sudoadduser new

sudousermod new -G four

sudoadduser oppo

sudousermod oppo -G four

sudoadduser vivo

sudousermod vivo -G four

sudoadduserchina

sudousermodchina -G four

cat /etc/group

cd EVERYONE/

cat /etc/passwd

ls

ls -l

su apple

cat /etc/group

cat /etc/passwd

chgrp two vivo

cd /home

ls

ls -l

cd ~

ls -l

cd EVERYONE

ls -l

sudo root

ls -l

sudo -s

cd ../

cd ~

pwd

ls -l

sudousermod oppo -g one sudousermod vivo -g two

cat /etc/group

pwd

cd /EVERYONE

cd EVERYONE

cat /etc/group

chgrp

chgrp oppo

chgrp 1001 oppo

pwd

cd ..

cat /etc/group

chgrp 1001 oppo

chgrp one oppo

cat /etc/group

whoami

ls

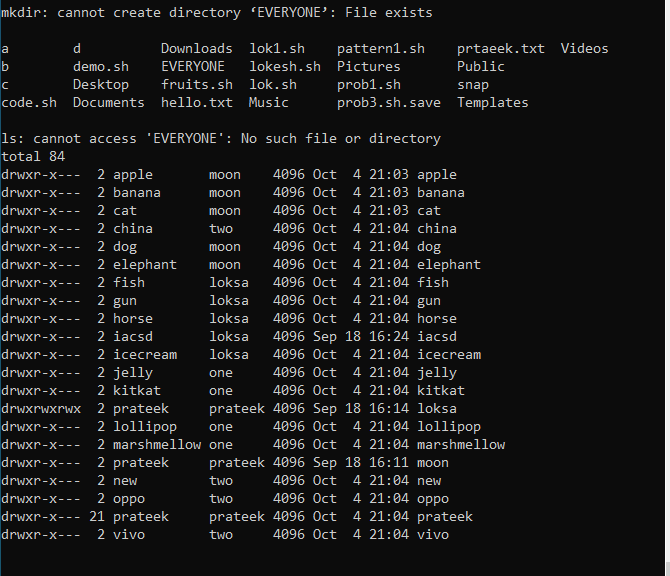
cd EVERYONE

ls

su oppo

cd ~

Pwd



# **ASSIGNMENT:5**

**﻿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((j=1;j<11;j++))

do

echo -n "$j ";

done

echo ""



**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

sum=0

for((j=1;j<11;j++))

do

echo -n "$j ";

sum=$(($sum+$j))

done

echo ""

echo "sum 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

read -p "enter no : " num

sum=0

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

do

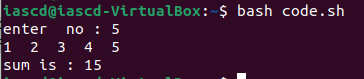
echo -n "$j ";

sum=$(($sum+$j))

done

echo ""

echo "sum is : $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

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

do

read -p "enter no : " num

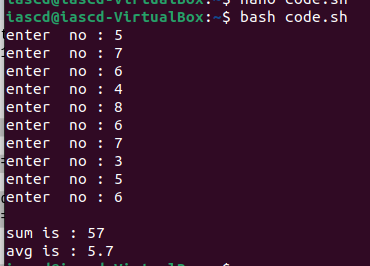
sum=$(($sum+$num))

done

echo ""

echo "sum is : $sum"

echo "avg is : $(($sum/10)).$(($sum%10))"



**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

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

do

read -p "enter no : " num

sum=1

for((j=1;j<=3;j++))

do

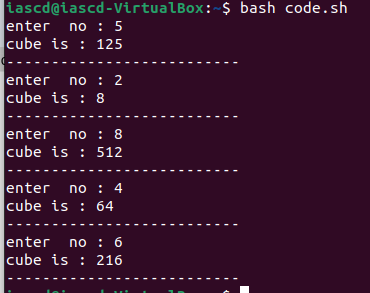
sum=$(($sum\*$num))

done

echo "cube is : $sum"

echo "--------------------------"

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**

**...**

**...**

**15 X 10 = 150**

#!/bin/bash

read -p "enter no : " num

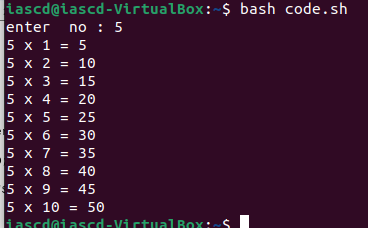
sum=1

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

do

echo "$num x $j = $(($j\*$num))"

done



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

read -p "enter no : " num

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

do

sum=1

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

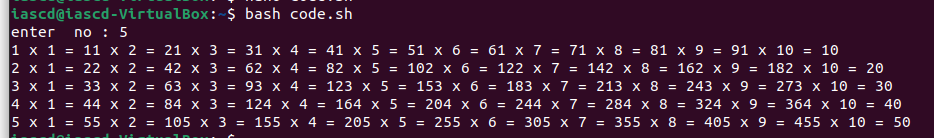
do

echo -n "$i x $j = $(($j\*$i))"

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 a number"

read n

echo "odd Numbers - "

i=1

k=0

sum=0

while [ $k -lt $n ]

do

rs=`expr $i % 2`

if [ $rs != 0 ]

then

echo -n " $i"

sum=$(($sum+i))

((k++))

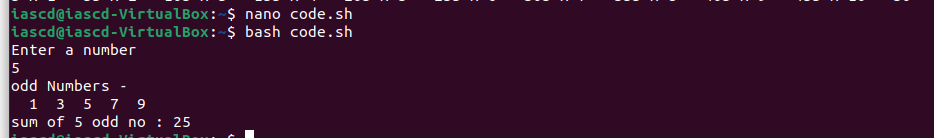
fi

((++i))

done

echo ""

echo "sum of $n odd no : $sum"



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

The pattern like :

\*

\*\*

\*\*\*

\*\*\*\*

#!/bin/bash

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

do

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

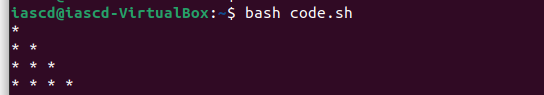
do

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

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

do

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

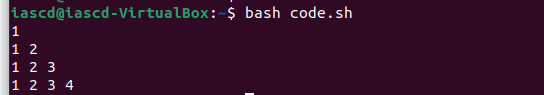
do

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

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

do

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

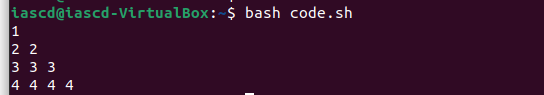
Do

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

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

do

for((j=1;j<=$i;j++,k++))

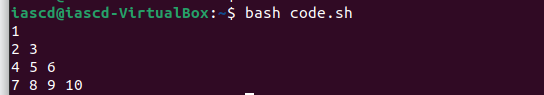
do

echo -n "$k "

done

echo ""

Done



**ASSIGNMENT:6**

**EXERCISE**

**1. Write a Shell Script to find maximum between two numbers.** #!\Bin\Bash

echo "Enter number 1 : "

read num1

echo "Enter number 2 : "

read num2

if [ $num1 -gt $num2 ]

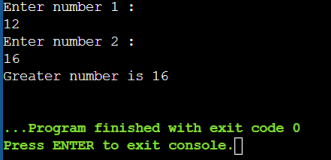
then

echo "Greater number is "$num1

else

echo "Greater number is "$num2

fi

**2. Write a Shell Script to find maximum between 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 "Greater number is "$num1

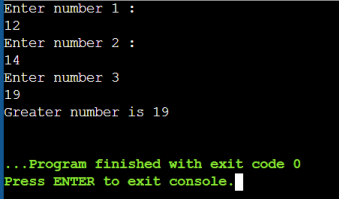
elif [ $num2 -gt $num1 ]&& [ $num2 -gt $num3 ] then

echo "Greater number is "$num2

else

echo "Greater number is "$num3

fi



**3. Write a Shell Script to check whether a number is negative, positive or zero.**

#!\Bin\Bash

echo "Enter number 1 : "

read num

if [ $num -lt0 ]

then

echo "Number is negative"

elif [ $num -gt0 ]

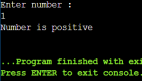
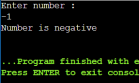
then

echo "Number is positive"

else

echo "Number is zero"

fi



**4. Write a Shell Script to check whether a number is divisible by 5 and 11 or not.**

#!\Bin\Bash

echo "Enter number : "

read num

if ((( $num % 5 == 0 ) && ( $num % 11 == 0 )))

then

echo "$num is divisible by both 5 and 11"

elif(( $num % 5 == 0 ))

then

echo "Number is divisible by 5"

elif(( $num % 11 ==0 ))

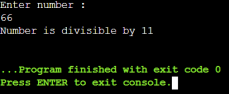
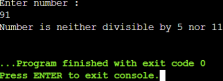
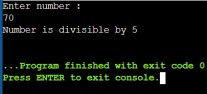
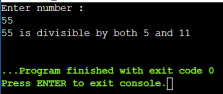
then

echo "Number is divisible by 11"

else

echo "Number is neither divisible by 5 nor 11"

fi



**5. Write a Shell Script to check whether a number is even or odd.** #!\Bin\Bash

echo "Enter number : "

read num

if (( $num % 2 == 0 ))

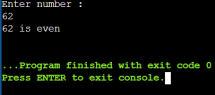
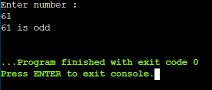
then

echo "$num is even"

else

echo "$num is odd"

fi



**6. Write a Shell Script to check whether a year is leap year or not.** #!\Bin\Bash

echo "Enter year : "

read num

y=$(( $num % 4 ))

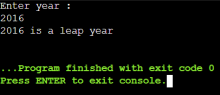
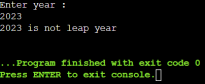
if (( $y != 0 )); then

echo "$num is not leap year"

else

echo "$num is a leap year"

fi



**7. Shell Script to print number between 1 to 10 in character format using switch-case.**

#!\Bin\Bash

echo "-------------MENU-------------"

echo "press number key to print"

echo "1 2 3 4 5 6 7 8 9 10 "

echo "Press 11 to Exit"

while [[ $choice != 11 ]]; do

echo "Enter your choice : "

read choice

case $choice in

1)

echo 'printed: 1'

;;

2)

echo 'printed: 2'

;;

3)

echo 'printed: 3'

;;

4)

echo 'printed: 4'

;;

5)

echo 'printed: 5'

;;

6)

echo 'printed: 6'

;;

7)

echo 'printed: 7'

;;

8)

echo 'printed: 8'

;;

9)

echo 'printed: 9'

;;

10)

echo 'printed: 10'

;;

11)

break;

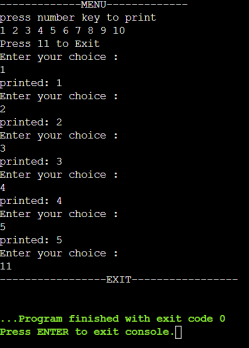
;;

\*)

echo "Invalid choice"

esac

done

echo "--------------EXIT---------"

**8. Shell Script to accept id from user to confirm department using switch-case.**

#!\Bin\Bash

echo "----------------------MENU---------------------"

echo -e

"\t\t1:Technical\n\t\t2:HR\n\t\t3:Sales\n\t\t4:Marketing\n\t\t5:Accountin g\n\t\t6:Production"

echo "Press 7 to Exit"

while [[ $choice != 7 ]]; do

echo -e "\nEnter your department id : "

read choice

case $choice in

1)

echo -e "\nTechnical department"

;;

2)

echo -e "\nHR department"

;;

3)

echo -e "\nSales department"

;;

4)

echo -e "\nMarketing department"

;;

5)

echo -e "\nAccounting department"

;;

6)

echo -e "\nProduction department"

;;

7)

break;

;;

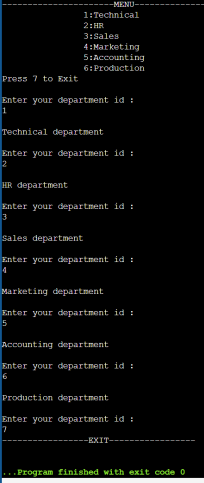
\*)

echo "Invalid choice"

esac

done

echo "-----------------EXIT-----------------"



**9. Shell Script to check password is correct or incorrect using switch case.**

**10. Shell Script to print day of week using switch-case.** #!\Bin\Bash

echo "-------------------MENU---------------------"

echo -e

"\t\t1:Monday\n\t\t2:Tuesday\n\t\t3:Wednesday\n\t\t4:Thursday\n\t\t5:Frid ay\n\t\t6:Saturday\n\t\t7:Sunday"

echo "Press 8 to Exit"

while [[ $choice != 8 ]]; do

echo -e "\nEnterday : "

read choice

case $choice in

1)

echo "Monday"

;;

2)

echo "Tuesday"

;;

3)

echo "Wednesday"

;;

4)

echo "Thursday"

;;

5)

echo "Friday"

;;

6)

echo "Saturday"

;;

7)

echo "Sunday"

;;

8)

break;

;;

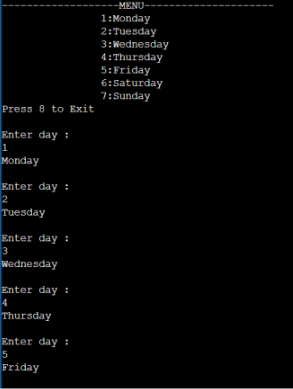
\*)

echo "Invalid choice"

esac

done

echo "-----------------EXIT-----------------"



**11. Shell Script to create calculator using switch-case.** #!\Bin\Bash

echo "-------------------MENU---------------------"

echo -e

"\t\t1:Sum\n\t\t2:Substract\n\t\t3:Multiply\n\t\t4:Divide\n\t\t5:Square\n \t\t6:Cube"

echo "Press 7 to Exit"

while [[ $choice != 7 ]]; do

echo -e "\nEnterchoice : "

read choice

case $choice in

1)

echo -e "\nEnter number 1 : "

read num1

echo "Enter number 2 : "

read num2

echo "Sum of $num1 and $num2 is " `expr $num1 + $num2` ;;

2)

echo -e "\nEnter number 1 : "

read num1

echo "Enter number 2 : "

read num2

echo "Substraction of $num1 and $num2 is " `expr $num1 - $num2` ;;

3)

echo -e "\nEnter number 1 : "

read num1

echo "Enter number 2 : "

read num2

echo "Multiply of $num1 and $num2 is " `expr $num1 \\* $num2` ;;

4)

echo -e "\nEnter number 1 : "

read num1

echo "Enter number 2 : "

read num2

echo "Division of $num1 and $num2 is " `expr $num1 / $num2` ;;

5)

echo -e "\nEnternumber : "

read num1

echo "Square of $num1 is " `expr $num1 \\* $num1`

;;

6)

echo -e "\nEnternumber : "

read num1

echo "Square of $num1 is " `expr $num1 \\* $num1 \\* $num1` ;;

7)

break;

;;

\*)

echo "Invalid choice"

esac

done

echo "-----------------EXIT-----------------"

