# Name:- Satyajit Hanmant Langote

# Class:- BCA-III SEM-V

# Roll No:-28

# LAB EXERCISE 1

**Copy, move Delete files from different directories.**

**mkdir command:** mkdir command is used to create the folder:

**Example:** $ mkdir Satyajit\_28

# Output:-

# 

**cd command :** cd command is used to change the folder:

**Example:** $ cd Satyajit\_28

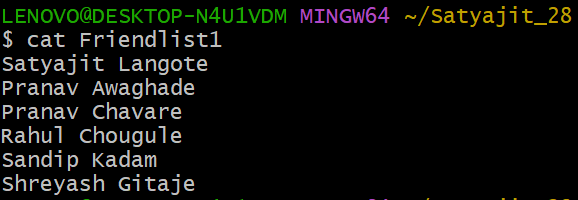
# Output:-

# 

**cat command :** cat command is used to create file, concatenate file, display contents of the file, Copy the file contents into new file, append the contents of File, to copy a file into the same directory**.**

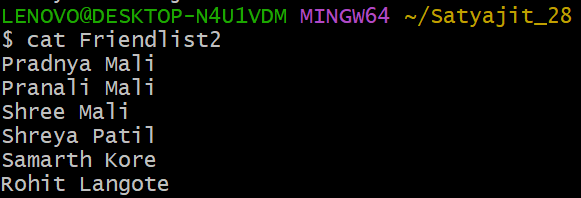
**Example:** $ cat > Friendlist1

**Output:-**

****

**Example:** $ cat > Friendlist2

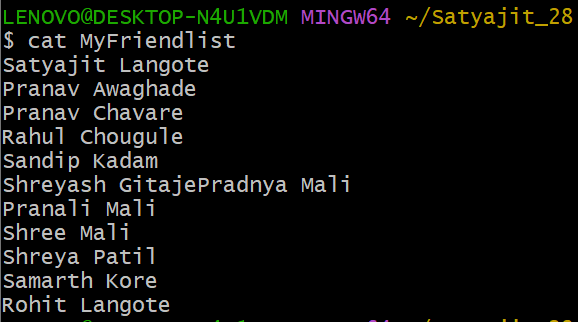
# Output:-

****

**Example:** $ cat Friendlist1 Friendlist2 > MyFriendlist

**Example:** $ cat MyFriendlist

# Output:-

****

**Example:** $ cat MyFriendlist > NewFriendlist

**Example:** $ cat NewFriendlist

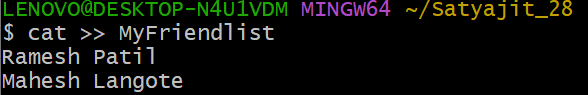
# Output:-

# 

**Cat >>:** To copy the all file content into new file

**Example:** $ cat >> MyFriendlist

# Output:-

****

**Example:** $ cat MyFriendlist

# Output:-

****

**cp command:** cp command is used to copy file into new file

**Example:** $ cp MyFriendlist CopyFriendlist

**Example:** $ cat CopyFriendlist

**Output:-**

****

**mv command:** mv command is used for rename the folder, copy the files one folder to another folder:

**Before Rename:-**

**Example:-** $ ls

# Output:-

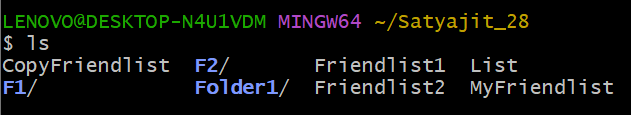
# 

**Example :-** $ mv NewFreindlist List

**After Rename:-**

**Example:-** $ ls

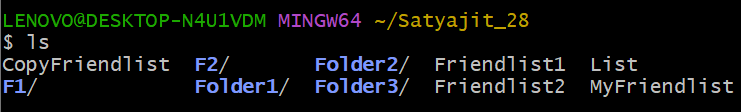
**Output:-**



**Example:** $ mkdir Folder2 Folder3

**Example:** $ ls

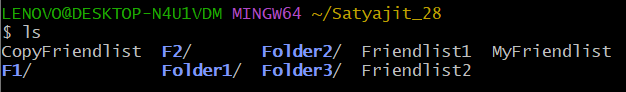
# Output:-



**Example:** $ mv List Folder1

**Example:** $ ls

# Output:-



**Example:** $ ls Folder1

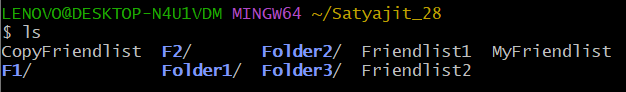
# Output:-



**rm command:** rm command is use for remove file from directory, delete files.

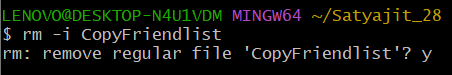
**Example:** $ ls

# Output:-



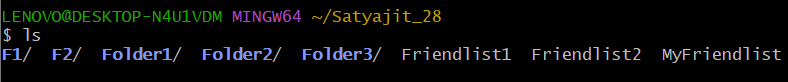
**Example:** $ rm -i CopyFriendlist

# Output:-



**Example:** $ ls

# Output:-



**Search the file using(\*):** The asterisk represents any number of unknown character. Use it when searching for documents or file for which you have only partial names.

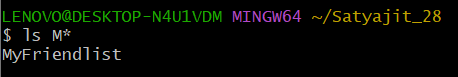
**Example:** $ ls F\*

**Output:-**

****

**Example:** $ ls M\*

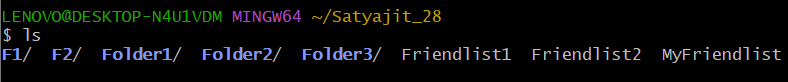
**Output:-**

****

**rmdir command:** rmdir command is used for delete folder from system:

**Example:** $ ls

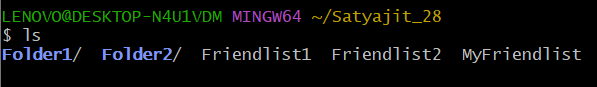
# Output:-





**Example:** $ ls

# Output:-



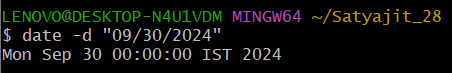
# LAB EXERCISE 2

# Basic Commands

**Date command:** date command is used for displaying dates, displaying the date in string format, displaying the past dates, displaying the future dates.

**Example:** $ date -d "09/30/2024"

# Output:-



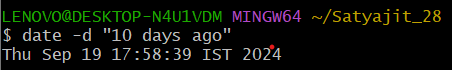
**Example:** $ date -d "2 years ago"

**Output:-**

****

**Example:** $ date -d "10 days ago"

# Output:-

****

**Example:** $ date -d "2 months ago"

# Output:-

# 

# 

**Example:** $ date --date "next tue"

# Output:-

# 

**Example:** $ date --date "tomorrow"

# Output:-

# 

**Example:** $ date "+%A %B %D"

# Output:-

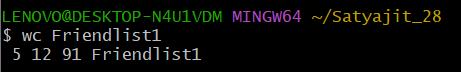
# 

# 

**wc command:** wc command stands for word count. As the name implies, it is mainly used for counting purpose and their options are used for counting lines, words and characters present in the file.

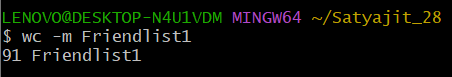
**Example:** $ wc Friendlist1

# Output:-

****

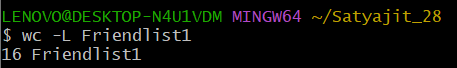
**Example:** $ wc -m Friendlist1

**Output:-**

****

**Example:** $ wc -L FriendList1

# Output:-

****

**Example:** $ ls MyFriendlist | wc –l

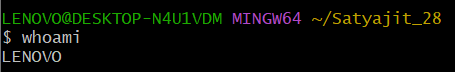
# Output:-

# 

**whoami command:** whoami command used for to display system’s username:

**Example:** $ whoami

# Output:-



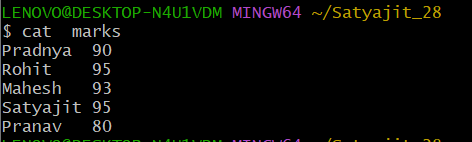
# LAB EXERCISE 3

# Use of filter commands

1. **cut command:** cut command is used for selecting specific column of a file. It is used to cut a specific sections by byte position, character and field.

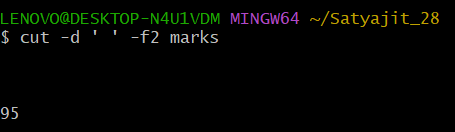
**Example:** $ cat marks

# Output:-

****

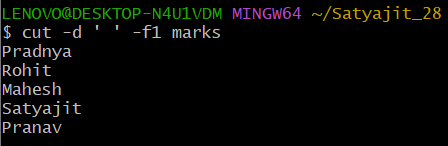
**Example:** $ cut -d ' ' -f2 marks

**Output:-**

****

**Example:** $ cut -d ' ' –f1 marks

**Output:-**

****

**Example:** $ cut -c 1,3 marks

# Output:-

# 

# 

**Example:** $ cut -c 1-3 marks

# Output:-

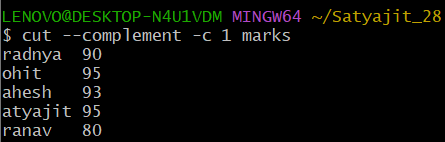
# 

# 

# 

**Example:** $ cut --complement -c 1 marks

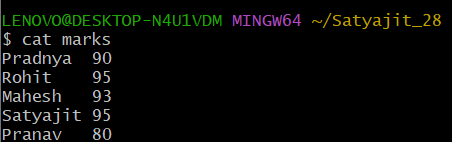
# Output:-

****

1. **paste command:** paste command is used for merge lines of a files horizontally. It consisting the sequentially corresponding lines of each file specified as an arguments, separated by tabs.

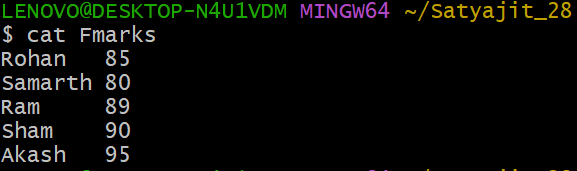
**Example:** $ cat marks

# Output:-

****

**Example:** $ cat Fmarks

**Output:-**

****

**Example:** $ paste marks Fmarks

# Output:-

# 

**Example:** $ paste -d '\_' marks Fmarks

# Output:-

# 

# 

# 

**Example:** $ paste -d '|' marks Fmarks

# Output:-

# 

# 

# 

1. **sort command:** sort command is used for sorting the file contents in an alphabetical and numerical order.

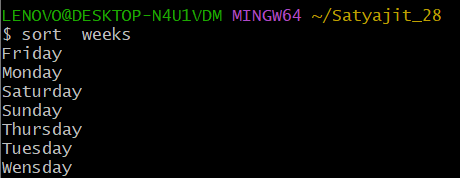
**Example:** $ cat weeks

# Output:-

# 

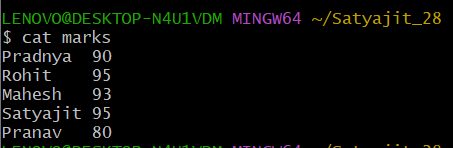
**Example:** $ sort weeks

# Output:-

****

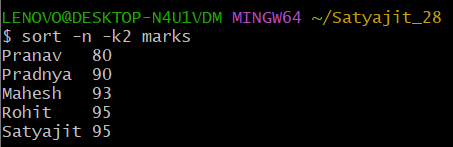
**Example:** $ cat marks

# Output:-



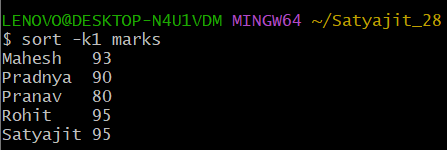
**Example:** $ sort -n -k2 marks

# Output:-



**Example:** $ sort –k1 marks

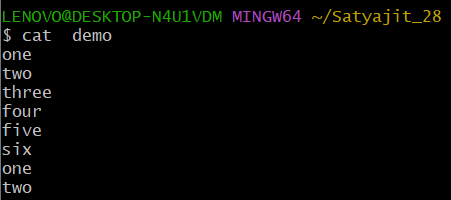
# Output:-



1. **uniq command:** uniq command is used to remove all repeated line from a file. Also, it can be used to display the count of words, only repeated lines, ignore characters, and compare specific field.

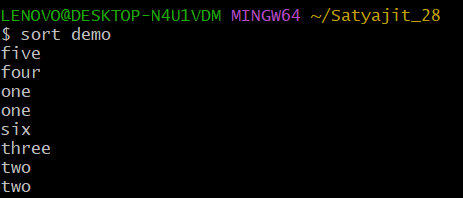
**Example:** $ cat demo

# Output:-

****

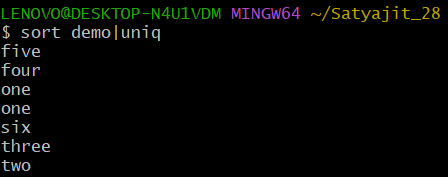
**Example:** $ sort demo

# Output:-

****

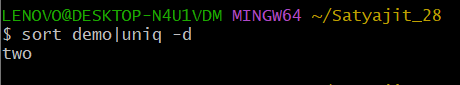
**Example:** $ sort demo | uniq

# Output:-



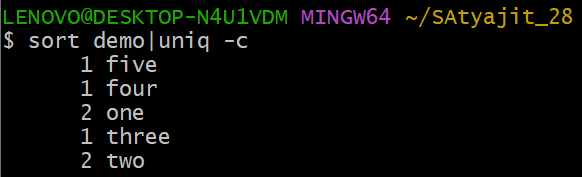
**Example:** $ sort demo | uniq –d

# Output:-



**Example:** $ sort demo | uniq –c

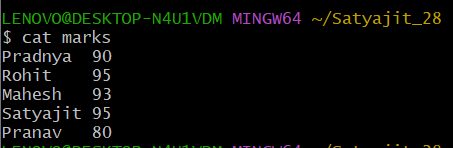
# Output:-



1. **grep command:** grep command stands for “global regular expression print”. grep command filters the content of a file which makes our search easy. grep command is generally used with pipe (|) and without pipe().

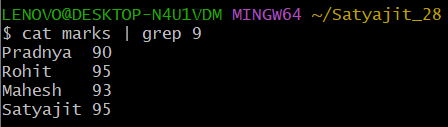
**Example:** $ cat marks

# Output:-



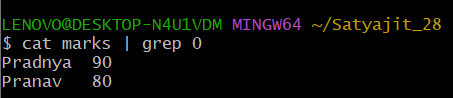
**Example:** $ cat marks | grep 9

# Output:-



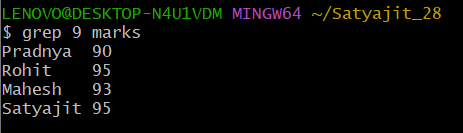
**Example:** $ cat marks | grep 0

# Output:-



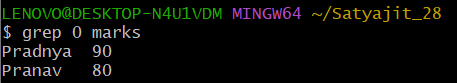
**Example:** $ grep 9 marks

# Output:-



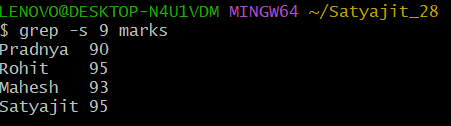
**Example:** $ grep 0 marks

# Output:-

****

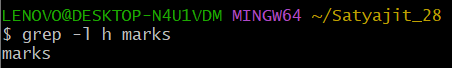
**Example:** $ grep -s 9 marks

# Output:-



**Example:** $ grep -l h marks

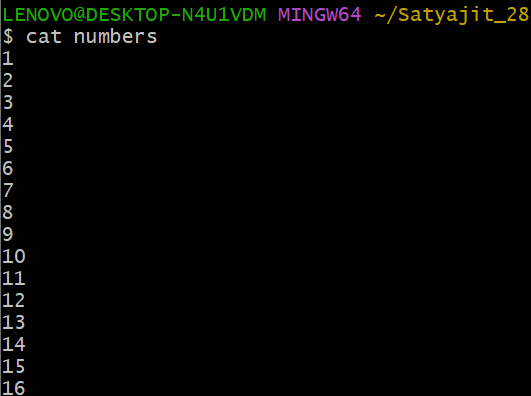
# Output:-



1. **head command**: head command displays the starting 10 lines of a file and display specified number of lines.

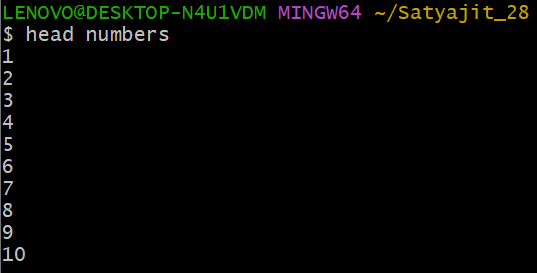
**Example:** $ cat numbers

# Output:-



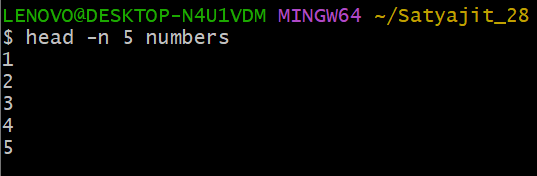
**Example:** $ head numbers

# Output:-



**Example:** $ head -n 5 numbers

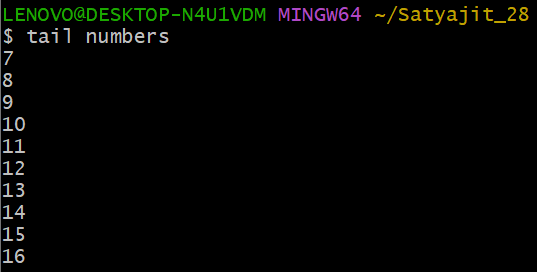
# Output:-



1. **tail command**: tail command displays the last 10 lines of a file and display specified number of lines.

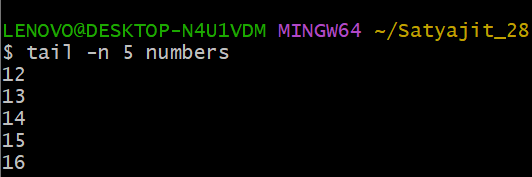
**Example:** $ tail numbers

# Output:-



**Example:** $ tail –n 5 numbers

# Output:-



**LAB EXERCISE 4**

**Write a shell script to demonstrate echo statement.**

**Example:**

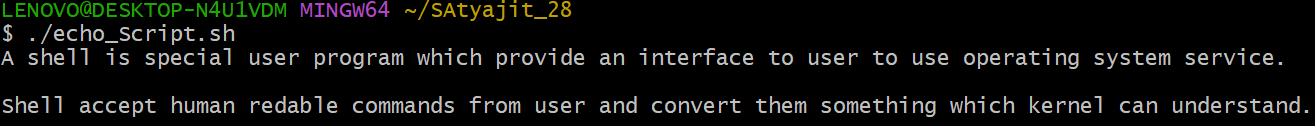
#!/bin/bash

echo A shell is special user program which provide an interface to user to use operating system service.

echo

echo Shell accept human readable commands from user and convert them into something which kernel can understand.

**Output:-**



**LAB EXERCISE 5**

**Write a shell script to demonstrate echo and read statement.**

**Example:**

#!/bin/bash

echo Enter your name?

read name

echo Enter your collage name

read c\_name

echo Enter your roll number

read roll\_no

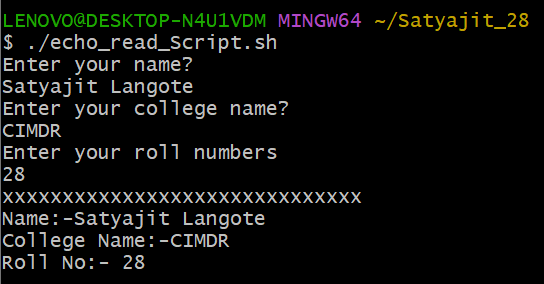
echo xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

echo Name: $name

echo Collage Name: $c\_name

echo Roll Number: $roll\_no

**Output:**



**LAB EXERCISE 6**

**Write a simple shell script to find exact number.**

**Example:**

#!/bin/bash

echo Enter Number

read num

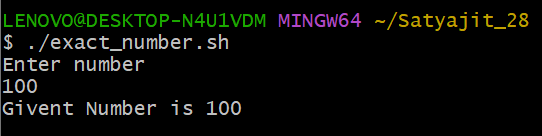
if [ $num == 100 ]

then

echo Given Number is 100

fi

**Output:**



**LAB EXERCISE 7**

**Write a shell script to find the candidate is eligible for voting or not.**

**Example:**

#!/bin/bash

echo Enter Age

read age

if [ $age -ge 18 ]

then

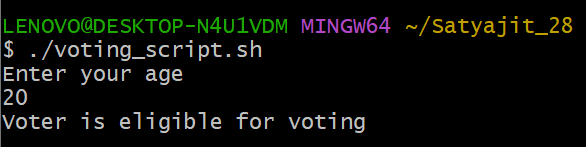
echo Voter is Eligible for voting

else

echo The voter is not eligible for voting

fi

**Output:**



**LAB EXERCISE 8**

**Write a shell script to find whether a given no is even or odd.**

**Example:**

#!/bin/bash

echo Enter number:

read num

rem=$(( $num % 2 ))

if [ $rem -eq 0 ]

then

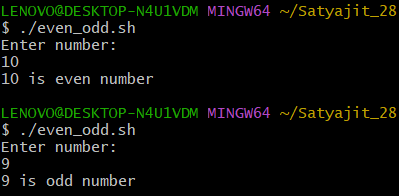
echo "$num is even number"

else

echo "$num is odd number"

fi

**Output:**



**LAB EXERCISE 9**

**Write a shell script WAP to find out maximum no from given three numbers.**

**Example:**

#!/bin/bash

echo Enter the first number

read num1

echo Enter the second number

read num2

echo Enter the third number

read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]

then

echo "$num1 is greater than $num2 and $num3"

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

then

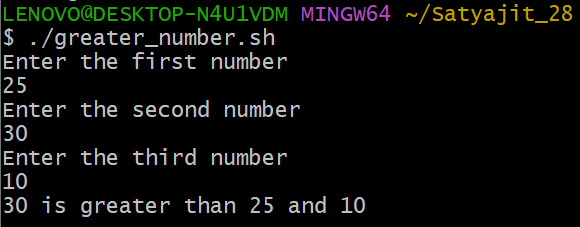
echo "$num2 is greater then $num1 and $num3"

else

echo "$num3 is greater than $num1 and $num2"

fi

**Output:**



**LAB EXERCISE 10**

**Write a shell script to calculate simple interest.**

**Example:**

#!/bin/bash

echo "Enter the principle amount:"

read price

echo "Enter the rate of interest:"

read rate

echo "Enter the number of years:"

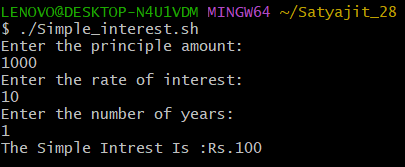
read years

i=`expr $price \\* $rate \\* $years`

k=` expr $i / 100 `

echo "The Simple Intrest Is :Rs.$k"

**Output:**



# LAB EXERCISE 11

**Write a shell script to create mark sheet.**

# Example:

#!/bin/bash

echo "Enter the student name" read name

echo "Enter the marks of Linux" read linux

echo "Enter the marks of Java" read java

echo "Enter the marks of DWDM" read dwdm

echo "Enter the marks of Account" read ac

echo

echo "Name: $name" echo "Subject Marks" echo "Linux: $linux" echo "Java: $java"

echo "DWDM: $dwdm" echo "Account: $ac" echo

sum=`expr $linux + $java + $dwdm + $ac` echo "The total sum is: $sum"

per=`expr $sum / 4` echo "Percentage: $per" echo

if [ $per -ge 75 ] then

echo "$name You get Distinction" elif [ $per -ge 60 ]

then

echo $name You get first class elif [ $per -ge 50 ]

then

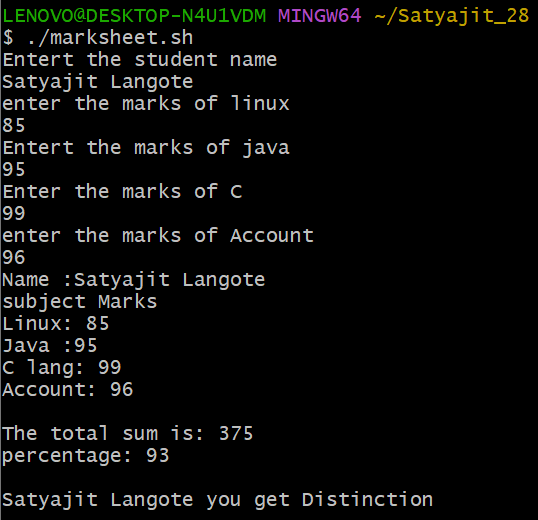
echo "$name You get second class" elif [ $per -ge 35 ]

then

echo $name You are pass else

echo You are fail fi

# Output:

****

**LAB EXERCISE 12**

**Write a shell script to use of switch case structure.**

# Example:

#!/bin/bash

echo "Enter the number: " read num

case $num in [0-9])

echo "You have entered a single digit number"

;;

[0-9][0-9])

echo "You have entered a two digit number"

;;

[0-9][0-9][0-9])

echo "You have entered a three digit number"

;;

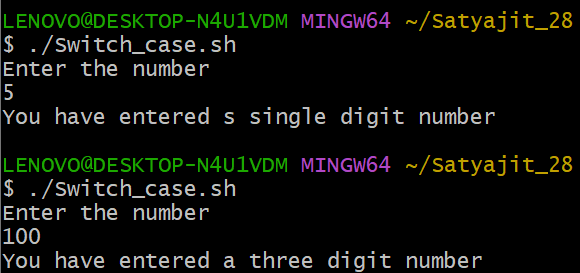
\*)

echo "Your entry does not match any of the conditions"

;;

esac

# Output:

****

**LAB EXERCISE 13**

**Write a shell script for simple for loop.**

# Example:

#!/bin/bash echo

echo "xxxxxxxxxxxxxxxxxxxx" for i in {1..5}

do

echo "$i" done echo

echo "xxxxxxxxxxxxxxxxxxxx" echo

for j in {11..20..2} do

echo "$j" done echo

echo "xxxxxxxxxxxxxxxxxxxx"

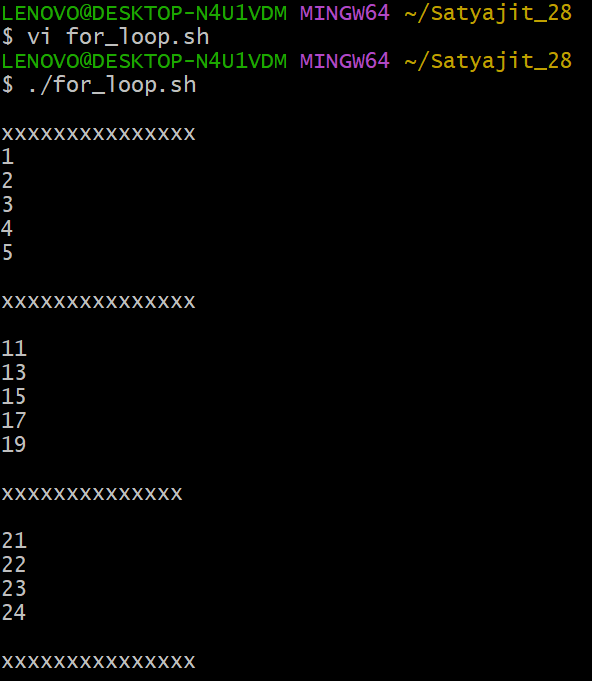
for (( k = 21; k < 25; k++ ))

do

echo "$k" done echo

echo "xxxxxxxxxxxxxxxxxxxx"

# Output:

****

**LAB EXERCISE 14**

**Write a shell script to display even numbers between 2 to 50 using while loop.**

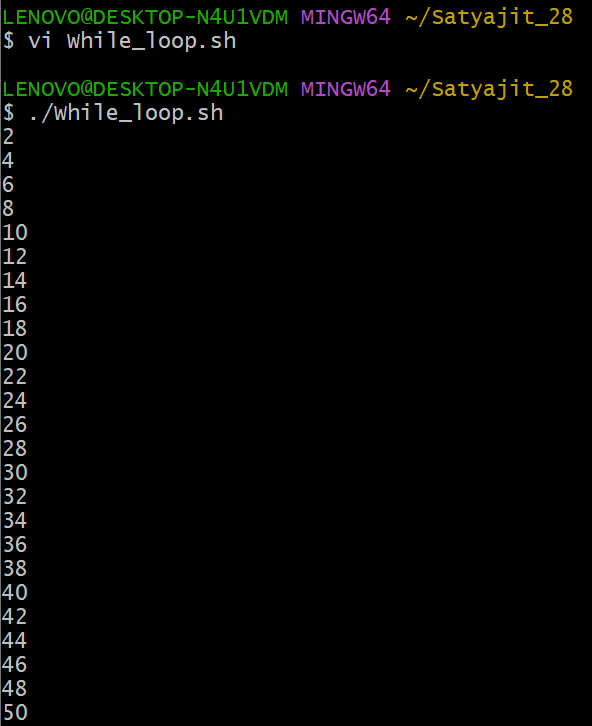
# Example:

#!/bin/bash n=2

while [ $n -le 50 ] do

echo $n n=$(( n+2 )) done

# Output:

****

**LAB EXERCISE 15**

**Write a shell script to find whether the given number is prime or not.**

# Example:

#!/bin/bash

echo Enter Number read num

for((i=2; i<=num/2; i++)) do

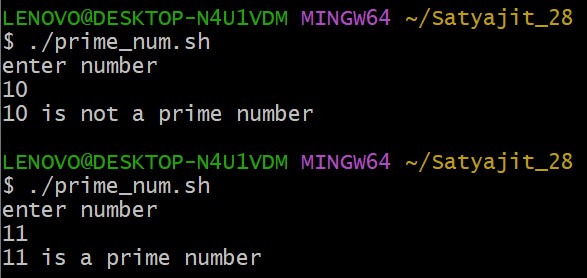
if [ $((num%i)) -eq 0 ] then

echo "$num is not a prime number." exit

fi done

echo "$num is a prime number."

# Output:

****

**LAB EXERCISE 16**

**Write a shell script to find whether the given number is prime or not.**

# Example:

#!/bin/bash

echo Enter Number read num

for((i=2; i<=num/2; i++)) do

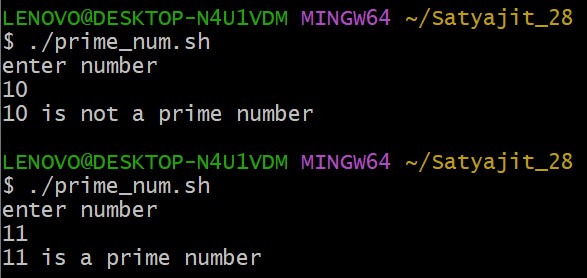
if [ $((num%i)) -eq 0 ] then

echo "$num is not a prime number." exit

fi done

echo "$num is a prime number."

# Output:

****

**LAB EXERCISE 17**

**Write a shell script to find whether the given number is perfect or not.**

# Example:

#!/bin/bash

echo Enter a number read no

i=1 ans=0

while [ $i -le `expr $no / 2` ] do

if [ `expr $no % $i` -eq 0 ] then

ans=`expr $ans + $i` fi

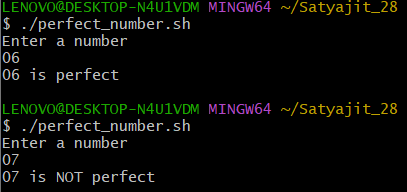
i=`expr $i + 1` done

if [ $no -eq $ans ] then

echo $no is perfect else

echo $no is NOT perfect fi

# Output: if [ expr $no / 2 ]; then

****

**LAB EXERCISE 18**

**Write a shell script to find whether the given string is palindrome or not.**

# Example:

#!/bin/bash

echo "Enter a String" read input reverse=""

len=${#input}

for (( i=$len-1; i>=0; i-- )) do

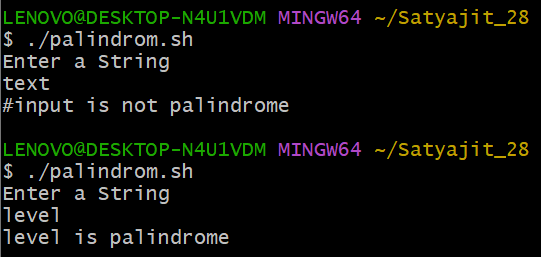
reverse="$reverse${input:$i:1}" done

if [ $input == $reverse ] then

echo "$input is palindrome" else

echo "$input is not palindrome" fi

# Output:

****

**LAB EXERCISE 19**

**Write a shell script whether Number is Armstrong number or not.**

# Example:

#!/bin/bash

echo "Enter a number: " read num

x=$num sum=0 r=0

n=0

while [ $x -gt 0 ] do

r=`expr $x % 10` n=`expr $r \\* $r \\* $r` sum=`expr $sum + $n` x=`expr $x / 10`

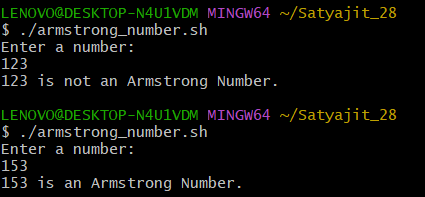
done

if [ $sum -eq $num ] then

echo "$num is an Armstrong Number." else

echo "$num is not an Armstrong Number." fi

**Output:**

****

**LAB EXERCISE 20**

**Write a shell script to find sum of digits of given Number using while.**

# Example:

#!/bin/bash

echo Enter Number read num

g=$num s=0

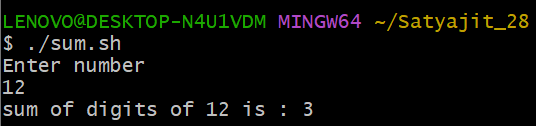
while [ $num -gt 0 ] do

k=$(( $num % 10 )) num=$(( $num / 10 )) s=$(( $s + $k ))

done

echo "sum of digits of $g is : $s"

# Output:

****

**LAB EXERCISE 21**

**Write a shell script to show the path and create directory.**

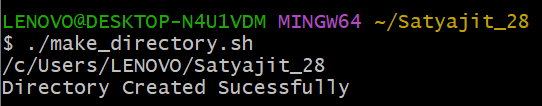
# Example:

#!/bin/bash pwd

mkdir Student

echo Directory Created Successfully

# Output:

****

**LAB EXERCISE 22**

**Write a shell script using for loop to display different commands.**

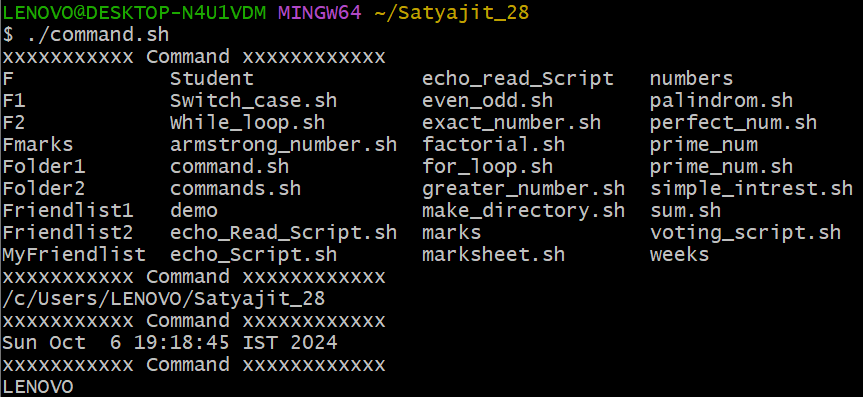
# Example:

#!/bin/bash

for command in ls pwd date whoami do

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Commands\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

$command done **Output:**



**LAB EXERCISE 23**

**Write a shell script that check whether the given string is found in a file or not. Display appropriate message.**

# Example:

#!/bin/bash

echo Enter file name read file

echo Enter string to find read string

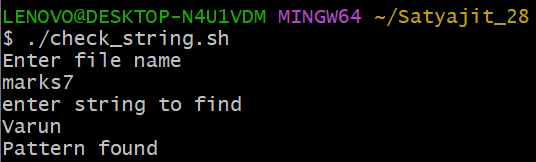
str=`grep $string $file` if [ $? -eq 0 ]

then

echo Pattern Found else

echo Pattern not found fi

# Output:

****

**LAB EXERCISE 24**

**Example:**

#!/bin/bash

echo ! ls command ! ls

echo !------------- cat command to show friendlist1 file !

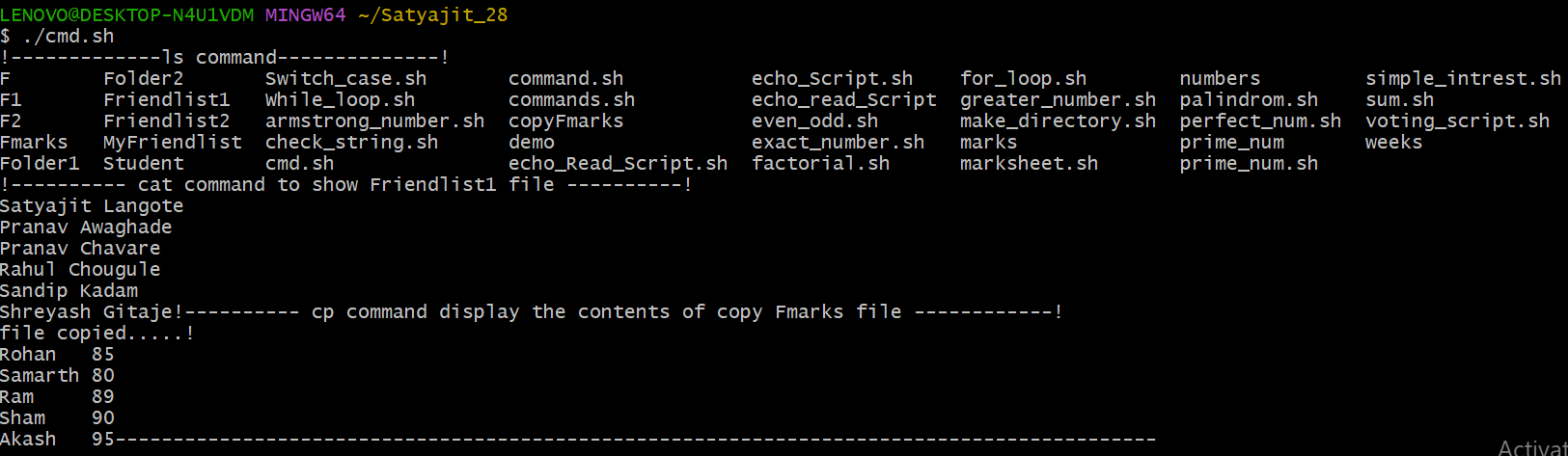
cat friendlist1

echo !------------- cp command display the contentens of copyFmarks file !

cp Fmarks copyFmarks echo file copied. !

cat copyFmarks echo

**Output:**

****