Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 1**

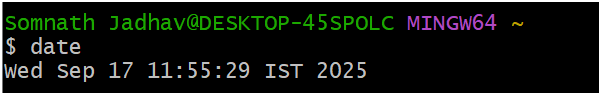
**Demonstrate the use of different GPU commands.**

**Q1: Date command: todays, tomorrows, 2years ago, 10 days ago, 2months ago, next Friday, Saturday 09/20/2025**

**Date Command:** Linux date command is used to display date, time zone etc. It also used to set date time of the linux system.

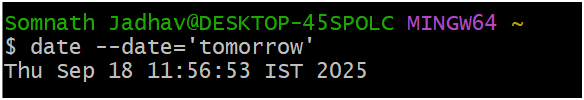
**Today:**

**Command:** $ date

****

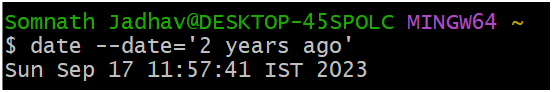
**Tomorrow:**

**Command:** $ date --date='tomorrow'

****

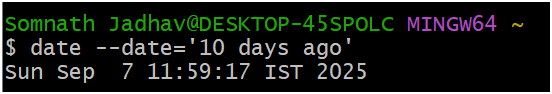
**2 years ago:**

**Command:** $ date --date='2 years ago'

****

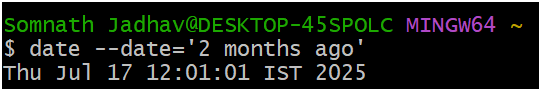
**10 days ago:**

**Command:** $ date --date='10 days ago'

****

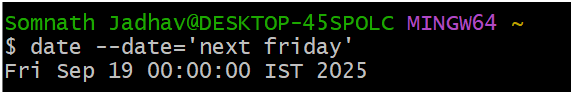
**2 months ago:**

**Command:** $ date --date='2 months ago'

****

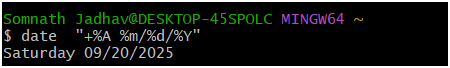
**Next Friday:**

**Command:** $ date --date='next friday'

****

**Saturday 09/20/2025:**

**Command:** $ date "+%A %m/%d/%Y"

****

**Q2. wc command: create file with employee add emp name,emp city and age. And perform the all options of wc command**

**wc command:** It stands for word count. It is used for counting purpose. Linux WC command helps in counting the lines, words & characters in the file

**Command:** $ cat > Employee

Somnath Sangli 20

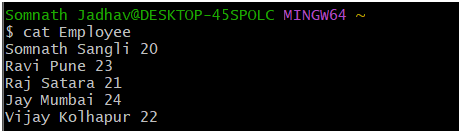
Ravi Pune 23

Raj Satara 21

Jay Mumbai 24

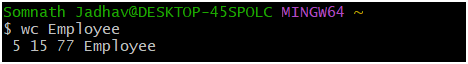
Vijay Kolhapur 22

**Command:** $ cat Employee

****

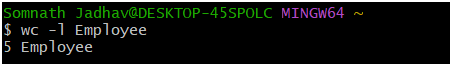
**Total Word Count:**

**Command:** $ wc Employee

****

**-l:**

**Command:** $ wc -l Employee

****

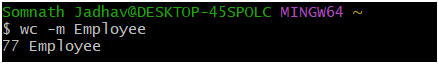
**-w:**

**Command:** $ wc -w Employee

****

**-m:**

**Command:** $ wc -m Employee

****

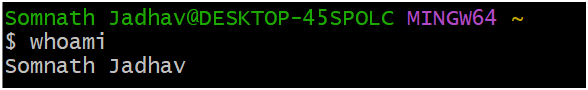
**-L:**

**Command:** $ wc -L Employee

****

**Command:** $ whoami

**whoami command:** This command is used to get information about currently logged-in user on the system

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

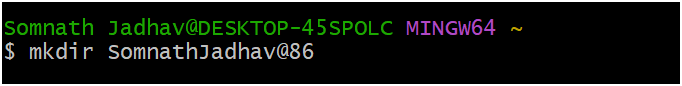
**LAB EXERCISE 2**

**Demonstrate the use of file handling commands and Directory handling commands**

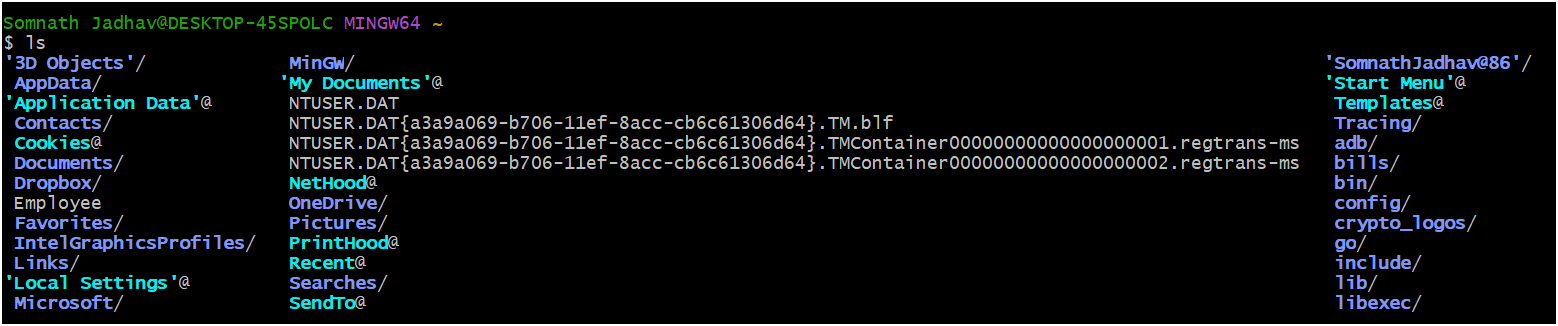
**Q1. Create folder with your name followed by roll no.**

**Command:** $ mkdir SomnathJadhav@86

**mkdir:** It stands for make directory. With the help of this command you can create a new directory where you want in a system

****

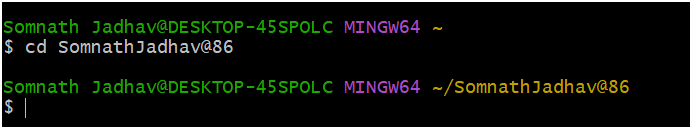
**Command: $ ls**

****

**Q2. Change the directory from home to your directory**

**Command:** $ cd SomnathJadhav@86

**cd:** This command is used to move from one directory to another directory

****

**Q3. Create two files in the same directory. (friendlist1 and friendlist2 – add any 5 records with name and surname)**

**Command**: $ cat > friendlist1

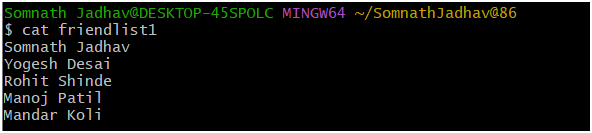
Somnath Jadhav

Yogesh Desai

Rohit Shinde

Manoj Patil

Mandar Koli

****

**Command:** $ cat > friendlist2

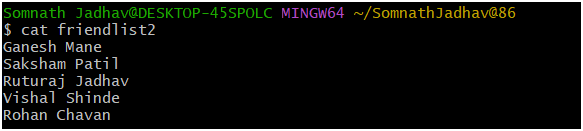
Ganesh Mane

Saksham Patil

Ruturaj Jadhav

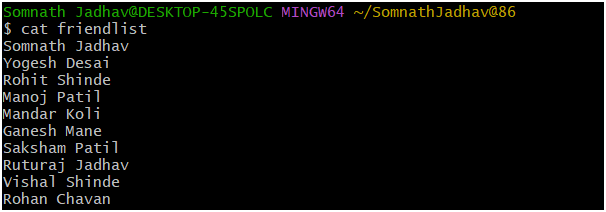
Vishal Shinde

Rohan Chavan

****

**Q4. Concatenate the friendlist1 and friendlist2 and display the output.**

**Command:** $ cat friendlist1 friendlist2 > friendlist

****

**Q5. Append the file friendlist1 (add any 4 records)**

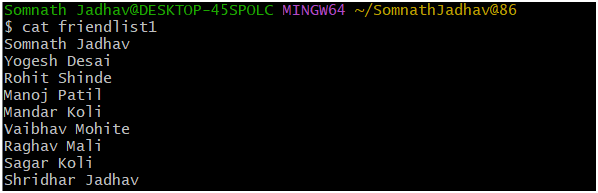
**Command:** $ cat >> friendlist1

Vaibhav Mohite

Raghav Mali

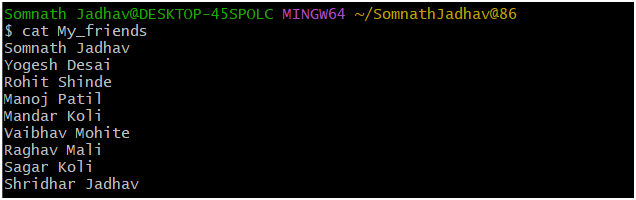
Sagar Koli

Shridhar Jadhav

****

**Q6. Copy the content of friendlist1 into new file My\_friends. (using cat command)**

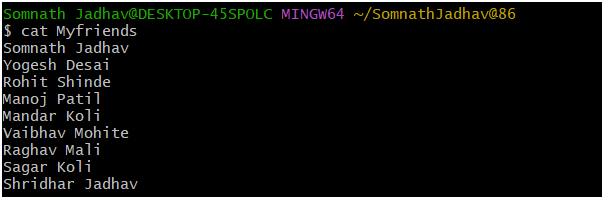
**Command:** $ cat friendlist1 > My\_friends

****

**Q7. Create the copy of My\_friends to Copy\_MyFriends (using cp command)**

**Cp**: cp stands for copy. This command is used to copy files or a group of files or directories. It creates an exact image of a file on a disk with with different filename.

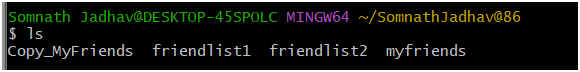
**Command:** $ cp My\_friends Myfriends

****

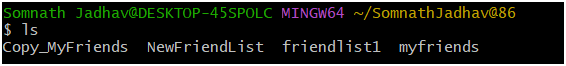
**Q8. Rename the file friendlist2 to NewFriendList.**

**mv** – Linux mv command is used to move existing file or directory from one location to another. It also used to rename a file or directory

**Before:**

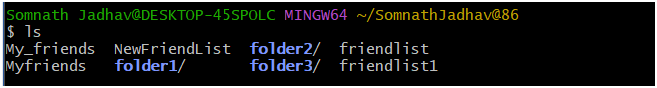


**Command:** $ mv friendlist2 NewFriendList

****

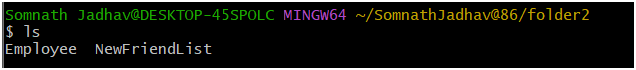
**Q9. Create the 3 folders folder1, folder2, folder3**

**Command:** $ mkdir folder1 folder2 folder3

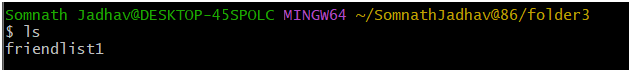
****

**Q10. Move the file employee, friendlist2 to folder2 and friendlist1 to folder3**

**Command:** $ mv Employee NewFriendList folder2

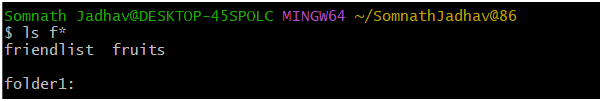
****

**Command:** $ mv friendlist1 folder3

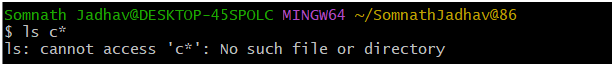
****

**Q11. Search the file and folder using asterisk whose name starts with f and c**

**Command:** $ ls f\*

****

**Command:** $ ls c\*

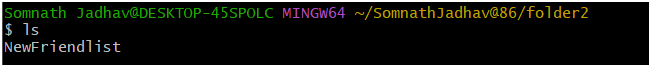
****

**Q12. Delete the employee from folder2 with permission.**

**Before:**

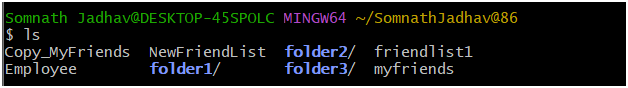
****

**Command:** $ rm -i employee

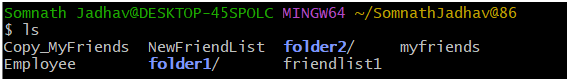
****

**Q13. Delete the folder3.**

**Before:**

****

**Command:** $ rm -r folder3

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 3**

**Demonstrate the use different filter commands**

**Q1. Create the file electronic\_products and add name and price of product**

**Command**: $ cat > electronic\_products

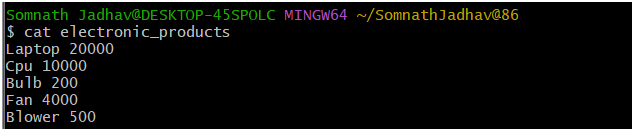
Laptop 20000

Cpu 10000

Bulb 200

Fan 4000

Blower 500

****

**Q2. Create the file stationary\_products and add name and price of products**

**Command:** $ cat > stationary\_products

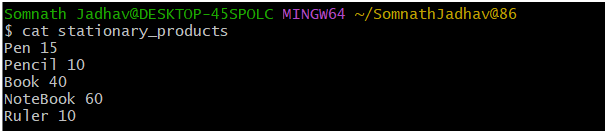
Pen 15

Pencil 10

Book 40

NoteBook 60

Ruler 10

****

**Q3. Create the file home\_products and add name and price of products**

**Command:** $ cat > home\_products

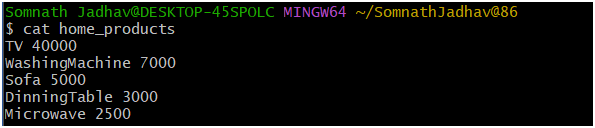
TV 40000

WashingMachine 7000

Sofa 5000

DinningTable 3000

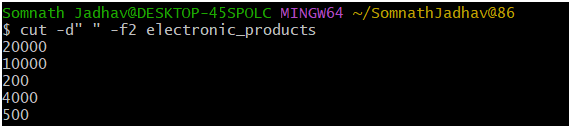
Microwave 2500

****

**Q4. Display the contents of column 2 from electronic\_products file**

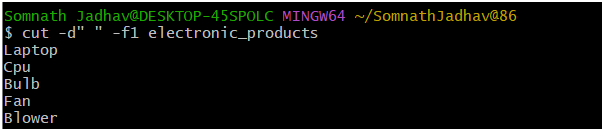
**Cut:** This command is used for selecting a specific column of a file. It is used to cut a specific section by byte, position, character & writes then to the standard output.

**Command:** $ cut -d" " -f2 electronic\_products

****

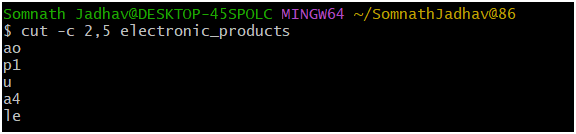
**Q5. Display the contents of column 1 from electronic\_products file**

**Command:** $ cut -d" " -f1 electronic\_products

****

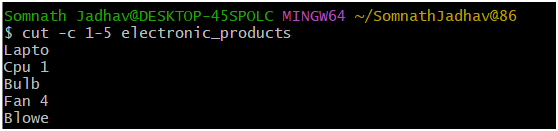
**Q6. Cut the characters from position 2 and 5**

**Command:** $ cut -c 2,5 electronic\_products

****

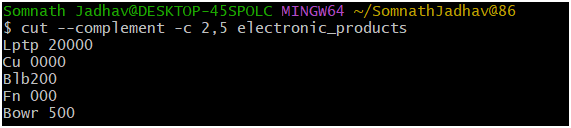
**Q7. Cut the characters form range 1-5**

**Command:** $ cut -c 1-5 electronic\_products

****

**Q8. Demonstrate the use of complement option with above same position and range.**

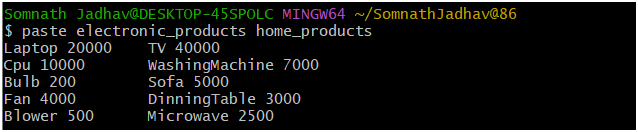
**Command:** $ cut --complement -c 2,5 electronic\_products

****

**Q9. Merge the both files horizontally.**

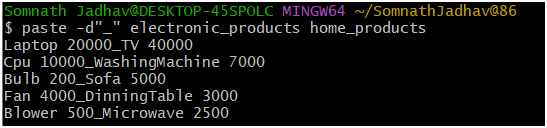
**Command:** $ paste electronic\_products home\_products

**Paste: Paste command allows you to merge lines of files horizontally. It outputs lines consisting of the sequentially corresponding lines of each file specified as an argument & separated by tabs.**

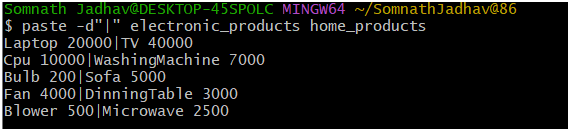
****

**Q10. Use \_ (underscore) and | delimiters while merge the files.**

**Command:** $ paste -d"\_" electronic\_products home\_products

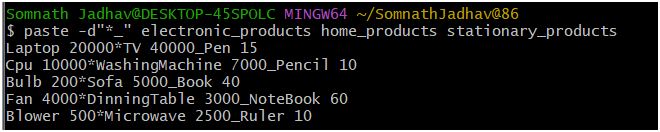
****

**Command:** $ paste -d"|" electronic\_products home\_products

****

**Q11. Merge above three files using different delimiters.**

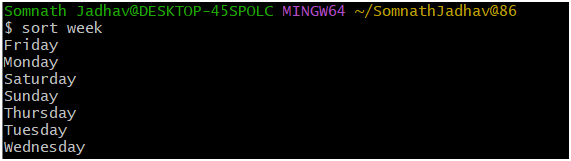
**Command:** $ paste -d"\*\_" electronic\_products home\_products stationary\_products

****

**Q12. Create the file week and sort it.**

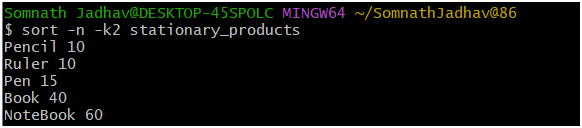
**Command:** $ sort week

**Sort-** Sort command is used to sort content of the file alphabetically.

****

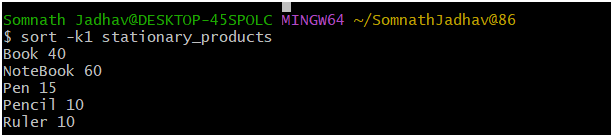
**Q13. Sort the stationary\_product file on the basis of product price.**

**Command:** $ sort -n -k2 stationary\_products

****

**Q14. Sort the stationary\_product file on the basis of column one.**

**Command:** $ sort -k1 stationary\_products

****

**Q15. Create the file fruits and add repeated data on that file.**

**Command:** $ cat > fruits

Orange

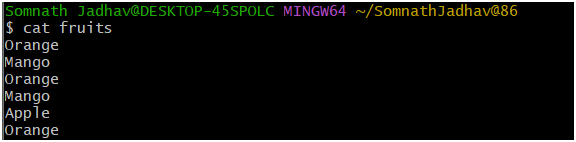
Mango

Orange

Mango

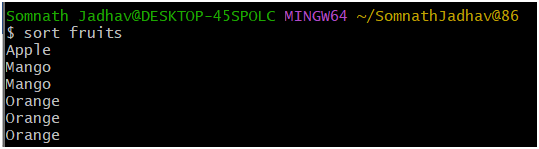
Apple

Orange

****

**Q16. Sort the file fruit.**

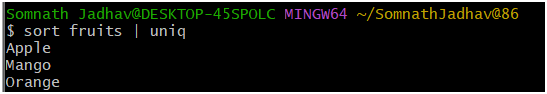
**Command:** $ sort fruits

****

**Q17. Remove the repeated lines from the file fruit.**

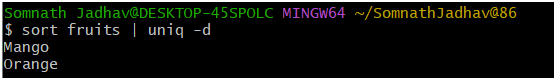
**Command**: $ sort fruits | uniq

**uniq –** Linux uniq command is used to remove all repeated lines from the file.

****

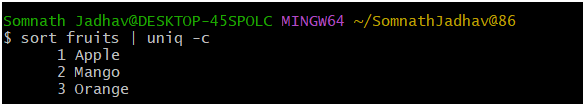
**Q18. Display the repeated lines from the file fruit.**

**Command:** $ sort fruits | uniq -d

****

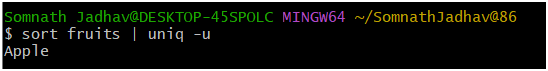
**Q19. Count the number of occurrences of words in the fruit file.**

**Command:** $ sort fruits | uniq -c

****

**Q20. Display the unique lines from fruit file.**

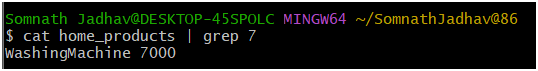
**Command:** $ sort fruits | uniq -u

****

**Q21. Display the home\_product whose price starts with 7 (use pipe symbol)**

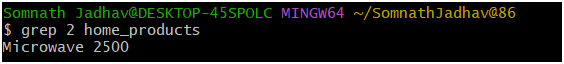
**Command:** $ cat home\_products | grep 7

**Grep:** grep is used for global regular expression print. Grep command filter the contents of a file which makes our search easy.

****

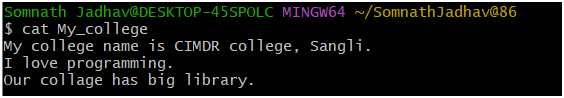
**Q22. Display the home\_product whose price start with 2 (without pipe symbol)**

**Command:** $ grep 2 home\_products

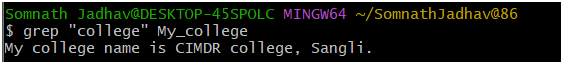
****

**Q23. Create file on My\_college. And find the line which contains word college using grep command.**

**Command:** cat My\_college

****

**Command**: $ grep "college" My\_college

****

**Q24. Create the file Numbers add one to fifteen numbers.**

**Command:** $ cat > Numbers

**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**10**

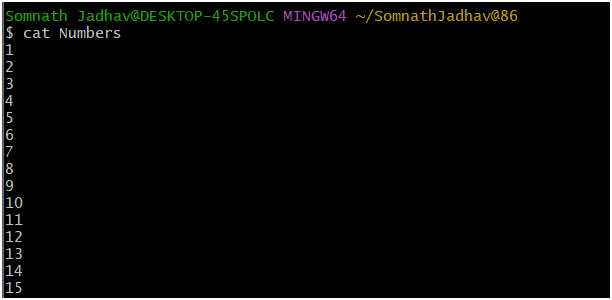
**11**

**12**

**13**

**14**

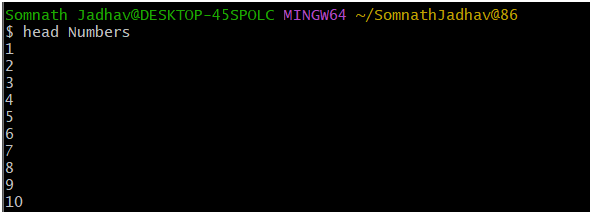
**15**

****

**Q25. Display the first 10 lines from numbers file**

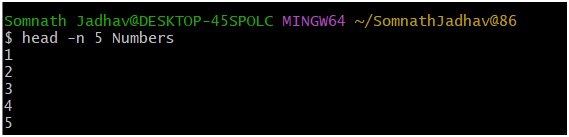
**Command**: $ head Numbers

**Head -** Head command in linux is used to display first lines of the file. By default, it displays first 10 lines of the file.

****

**Q26. Display the first 5 lines from file.**

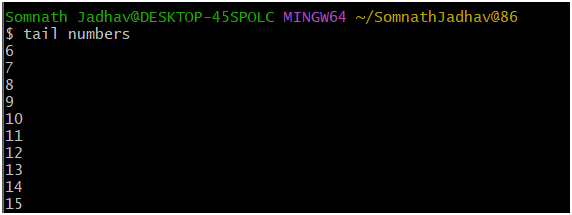
**Command:** $ head -n 5 Numbers

****

**Q27. Display the last 10 lines from numbers file.**

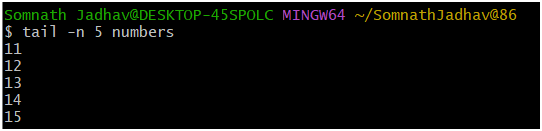
**Command:** $ tail numbers

**Tail-** Tail command in linux is used to display last lines of the file. By default, it displays last10 lines of the file.

****

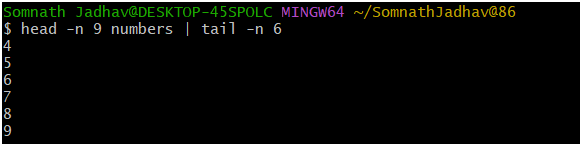
**Q28. Display the last 5 lines from number file.**

**Command:** $ tail -n 5 numbers

****

**Q29. Display the last 6 lines from first 9 lines from number file.**

**Command:** $ head -n 9 numbers | tail -n 6

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 4**

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

**Script:**

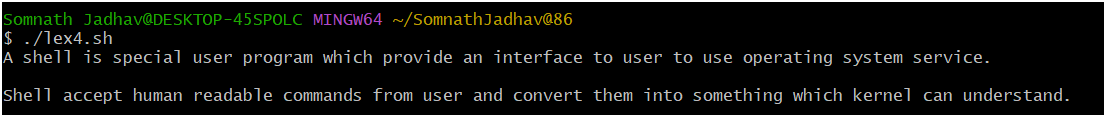
#!/bin/bash

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:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 5**

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

**Script:**

#!/bin/bash

read -p "Enter your name : " name

read -p "Enter your collage name : " c\_name

read -p "Enter your roll number : "

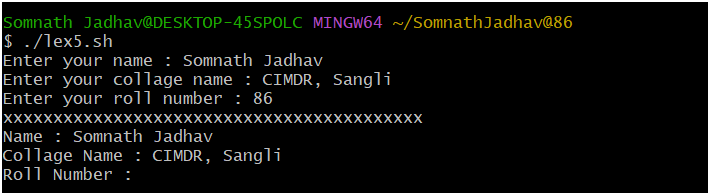
echo xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

echo Name : $name

echo Collage Name : $c\_name

echo Roll Number : $roll\_no

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 6**

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

**Script:**

#!/bin/bash

read -p "Enter Number : " num

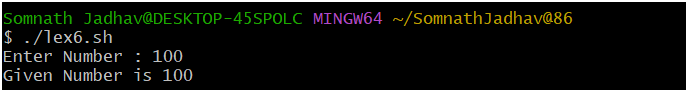
if [ $num == 100 ]

then

echo Given Number is 100

fi

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 7**

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

**Script:**

#!/bin/bash

read -p "Enter Age : " age

if [ $age -ge 18 ]

then

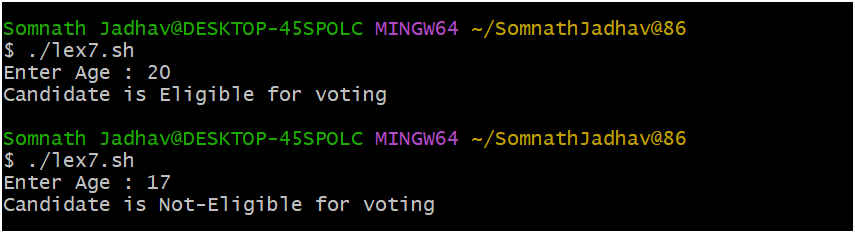
echo Candidate is Eligible for voting

else

echo Candidate is Not-Eligible for voting

fi

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 8**

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

**Script:**

#!/bin/bash

read -p "Enter number : " num

if [ $(($num % 2)) -eq 0 ]

then

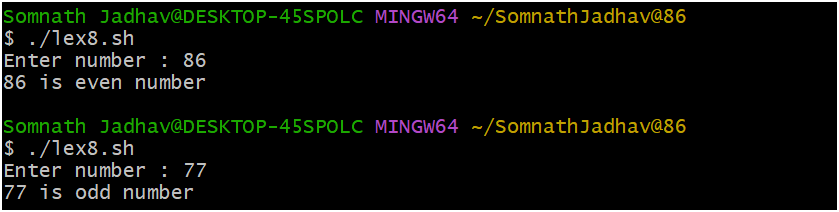
echo "$num is even number"

else

echo "$num is odd number"

fi

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 9**

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

**Script:**

#!/bin/bash

read -p "Enter the first number : " num1

read -p "Enter the second number : " num2

read -p "Enter the third number : " 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

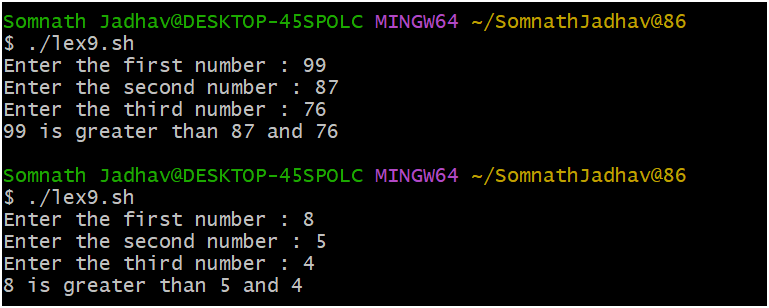
echo "$num2 is greater then $num1 and $num3"

else

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

fi

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 10**

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

**Script:**

#!/bin/bash

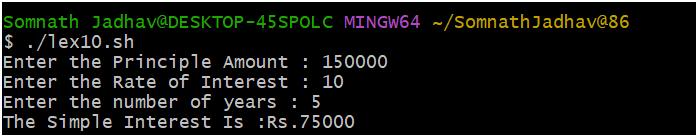
read -p "Enter the Principle Amount : " price

read -p "Enter the Rate of Interest : " rate

read -p "Enter the number of years : " years

echo "The Simple Interest Is :Rs.$(expr $price \\* $rate \\* $years / 100)"

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 11**

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

**Script:**

#!/bin/bash

read -p "Enter the Student Name : " name

read -p "Enter the Marks of Linux : " linux

read -p "Enter the Marks of Java : " java

read -p "Enter the Marks of DWDM : " dwdm

read -p "Enter the marks of Account : " ac

echo -e "-- \n Student Information--"

echo -e "\nName: $name"

echo -e "Subject Marks \n Linux: $linux \n Java: $java \n DWDM: $dwdm \n Account: $ac"

echo

total=$(expr $linux + $java + $dwdm + $ac)

echo "The total marks: $total"

per=$(expr $total / 4)

echo "Percentage: $per"

echo

if [ $per -ge 75 ]

then

echo "$name You got Distinction"

elif [ $per -ge 60 ]

then

echo $name You got first class

elif [ $per -ge 50 ]

then

echo "$name You got second class"

elif [ $per -ge 35 ]

then

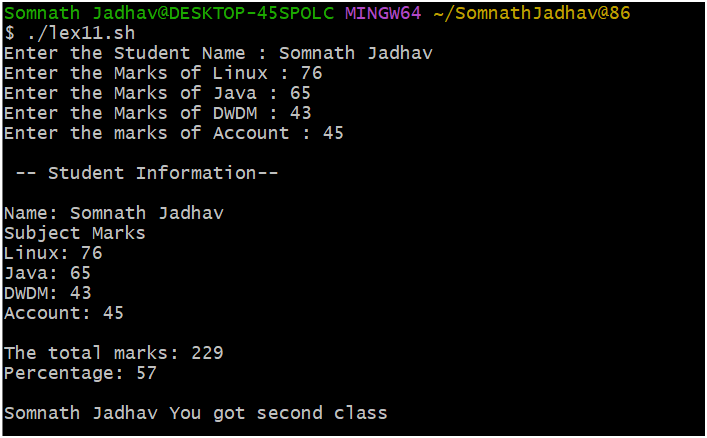
echo $name You are pass

else

echo You are fail

fi

**Output:**



Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 12**

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

**Script:**

#!/bin/bash

read -p "Enter the number : " 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"

;;

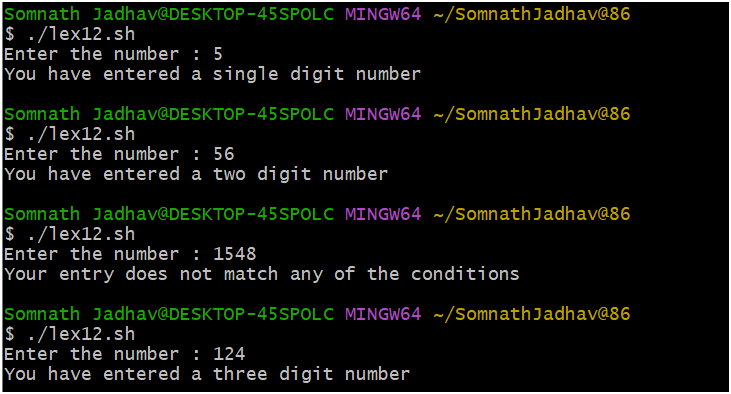
\*)

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

;;

esac

**Output:**



Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 13**

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

**Script:**

#!/bin/bash

for i in {1..5}

do

echo "$i"

done

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

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

do

echo "$j"

done

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

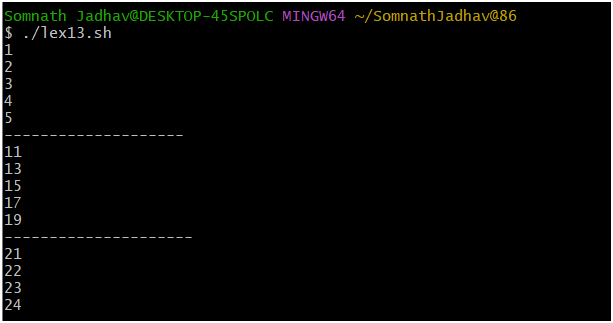
for (( k = 21; k < 25; k++ ))

do

echo "$k"

done

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 14**

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

**Script:**

#!/bin/bash

n=2

while [ $n -le 50 ]

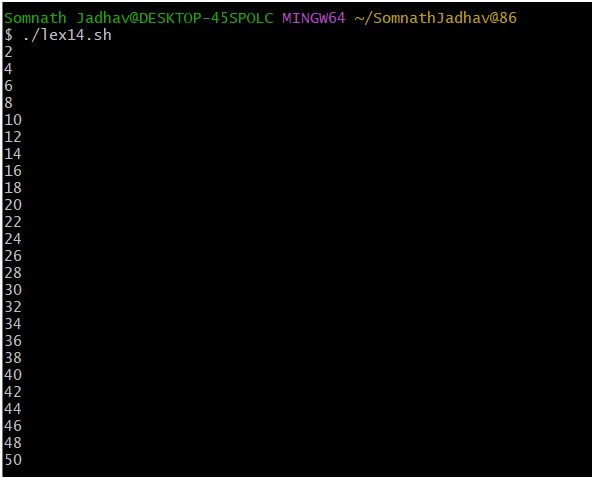
do

echo "$n "

n=$(( n+2 ))

done

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 15**

**Q. Write a shell script to calculate factorial of given number.**

**Script:**

#!/bin/bash

read -p "Enter an Number : " num

fact=1

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

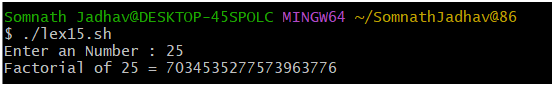
{

fact=$(( fact\*i ))

}

echo Factorial of $num = $fact

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 16**

**Q. Write a shell script to calculate factorial of given number.**

**Script:**

#!/bin/bash

read -p "Enter Number : " num

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

do

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

then

echo "$num Not prime number."

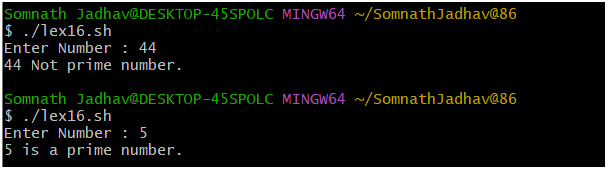
exit

fi

done

echo "$num is a prime number."

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 17**

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

**Script:**

#!/bin/bash

read -p "Enter a number : " 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

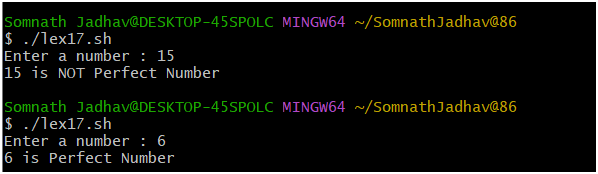
echo $no is Perfect Number

else

echo $no is NOT Perfect Number

fi

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 18**

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

**Script:**

#!/bin/bash

read -p "Enter a String : " input

reverse=""

len=${#input}

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

do

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

done

if [ $input == $reverse ]

then

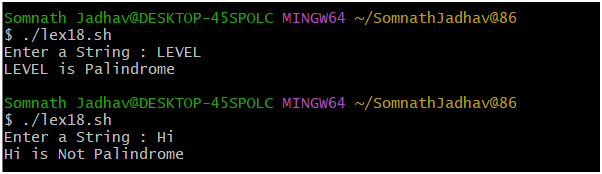
echo "$input is Palindrome"

else

echo "$input is Not Palindrome"

fi

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 19**

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

**Script:**

#!/bin/bash

read -p "Enter a number: " 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

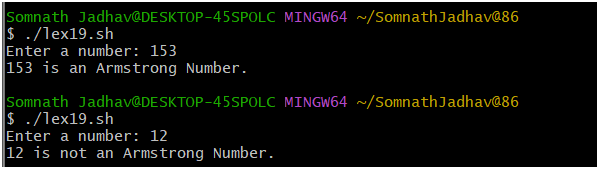
echo "$num is an Armstrong Number."

else

echo "$num is not an Armstrong Number."

Fi

**Output:**



Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 20**

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

**Script:**

#!/bin/bash

read -p "Enter Number : " num

g=$num

s=0

while [ $num -gt 0 ]

do

k=$(( $num % 10 ))

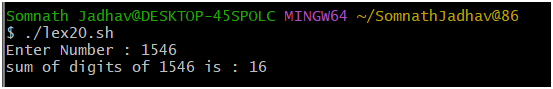
num=$(( $num / 10 ))

s=$(( $s + $k ))

done

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

**Output:**



Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 21**

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

**Script:**

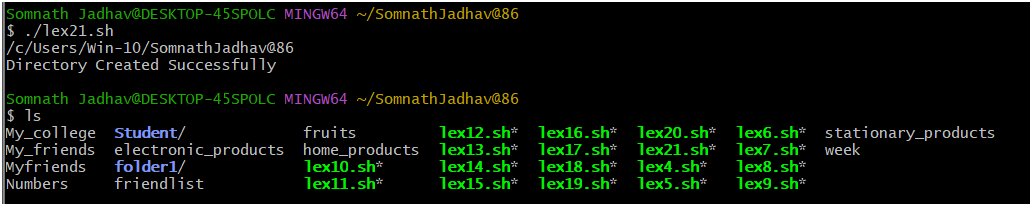
#!/bin/bash

pwd

mkdir Student

echo Directory Created Successfully

**Output:**



Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 22**

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

**Script:**

#!/bin/bash

for command in ls pwd date whoami

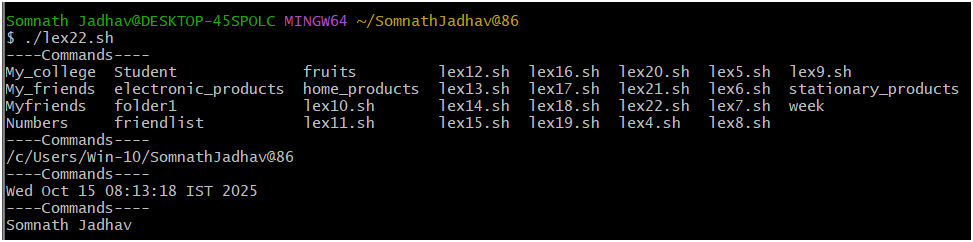
do

echo ----Commands----

$command

done

**Output:**



Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 23**

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

**Script:**

#!/bin/bash

read -p "Enter File Name : " file

read -p "Enter string to find : " string

str=`grep $string $file`

if [ $? -eq 0 ]

then

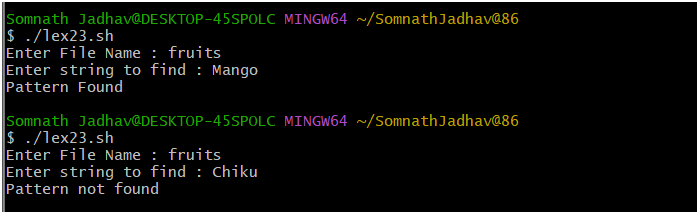
echo Pattern Found

else

echo Pattern not found

fi

**Output:**



Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 24**

**Q. Demonstrate cat and cp commands using shell script**

**Script:**

#!/bin/bash

echo -ls command -

ls

echo -cat command to show friendlist file -

cat friendlist

echo -cp command copy the contents of fruits file -

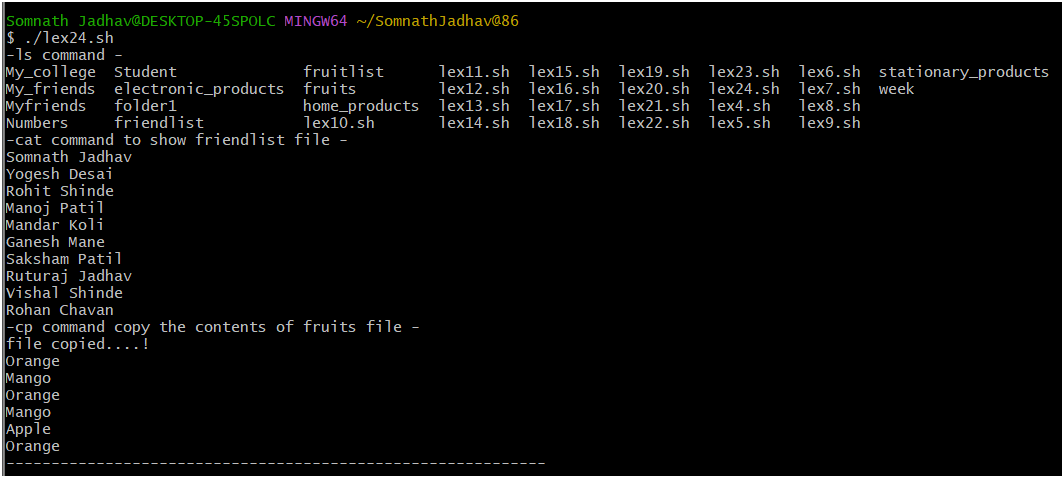
cp fruits fruitlist

echo file copied....!

cat fruitlist

echo ------------------------------------------------------------

**Output:**

****

Name: Jadhav Somnath Pandurang

Class: BCA – III Sem – V

Roll No: 86

**LAB EXERCISE 25**

**Q. Use different shell commands using case statements**

**Script:**

#!/bin/bash

echo -e "a) Show current date \nb) Show current directory \nc) List Files"

read -p "Enter your choice : " choice

case $choice in

a)

echo "Todays date is : $(date)"

;;

b)

echo "Your are in $(pwd)"

;;

c)

echo "Files in this directory"

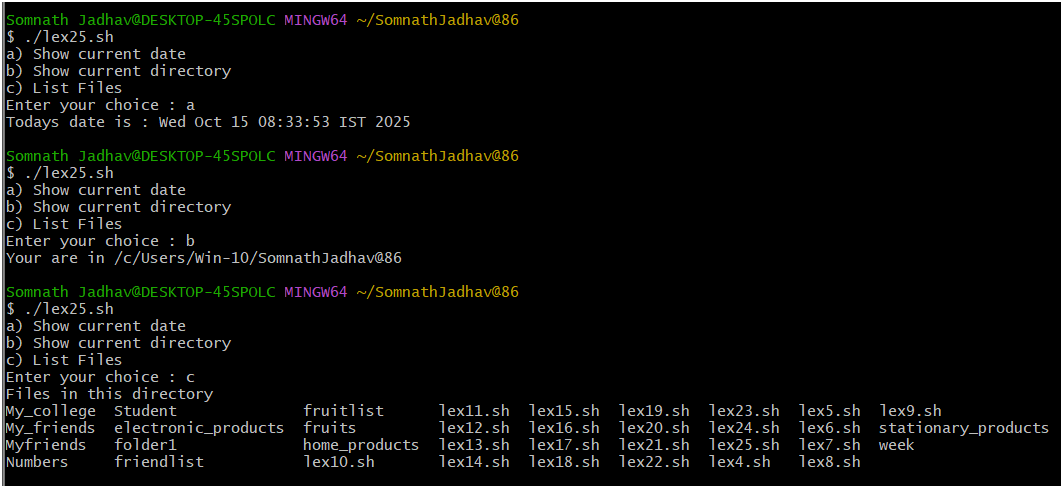
ls;;

\*)

echo "Invalid Option"

esac

**Output:**

****

**@@@**