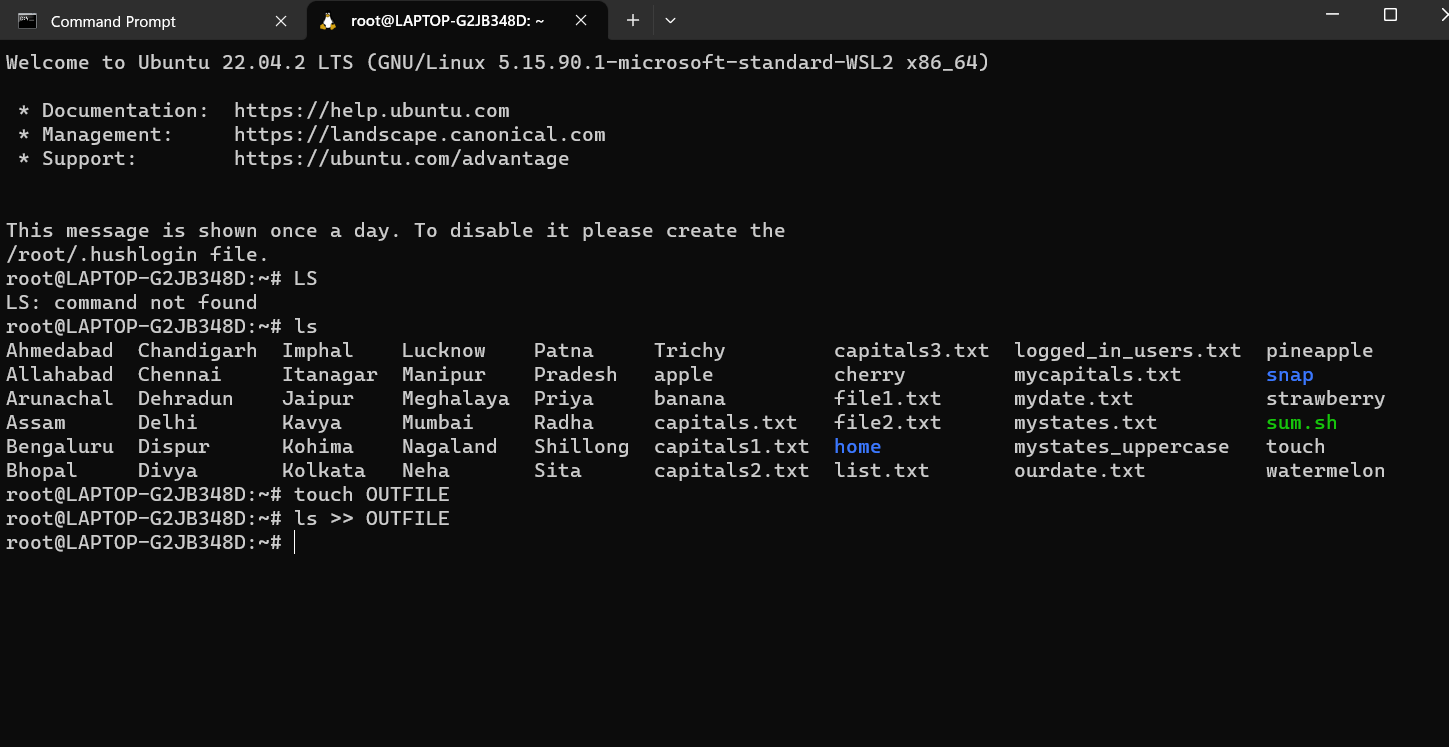
Name: Revati Shimpi

SY-D B2-11

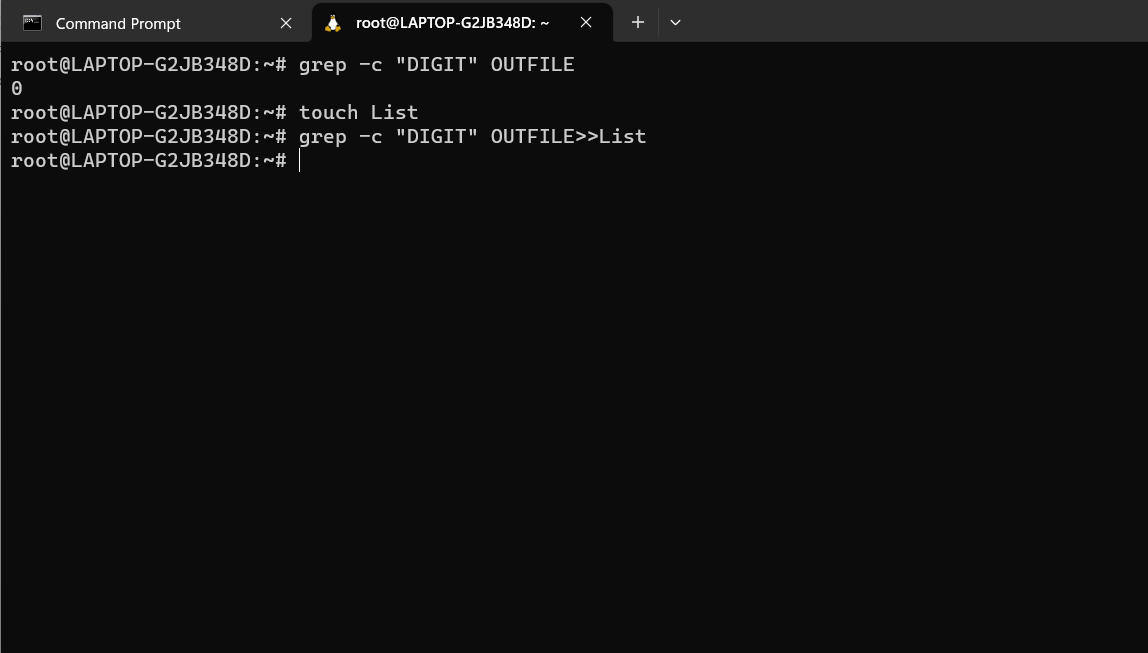
12210053

LINUX

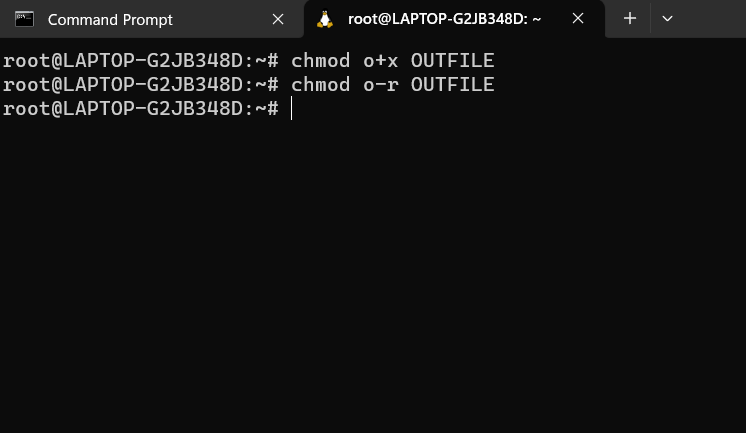
Problem Statement1) Redirect output of ls command to a file named OUTFILE.



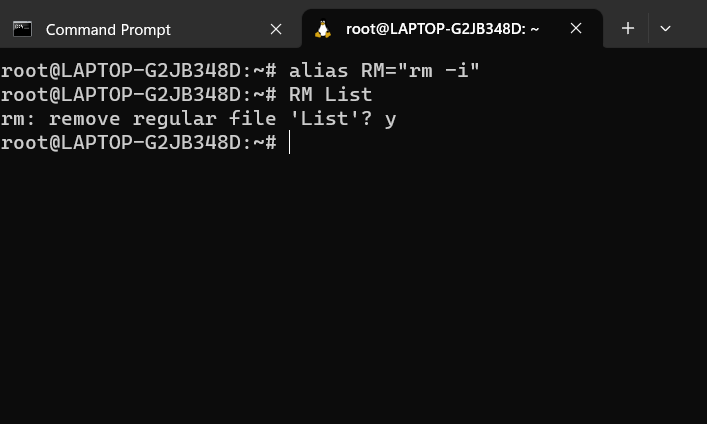
Problem Statement 2) To select the lines in file which has DIGIT as one of the character in that line and redirect the output to file named as LIST.



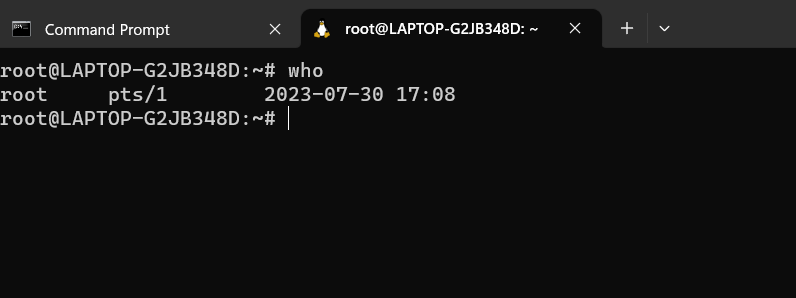
Problem Statement 3) Assign execute permission to owner and rmv read permission.



Problem Statement 4) Create an alias named as RM that always delete the file interactively.



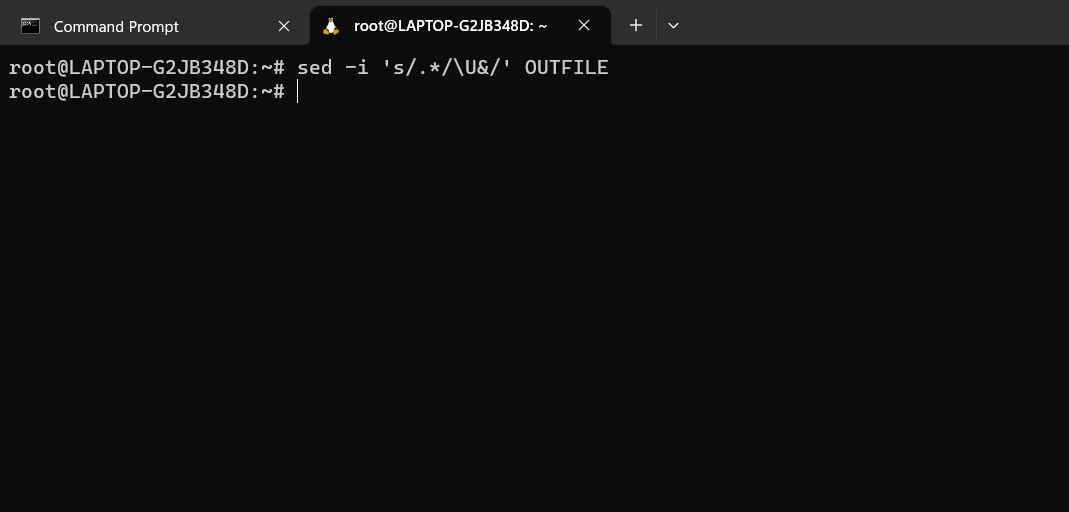
Problem Statement 5) Count currently logged-in users.



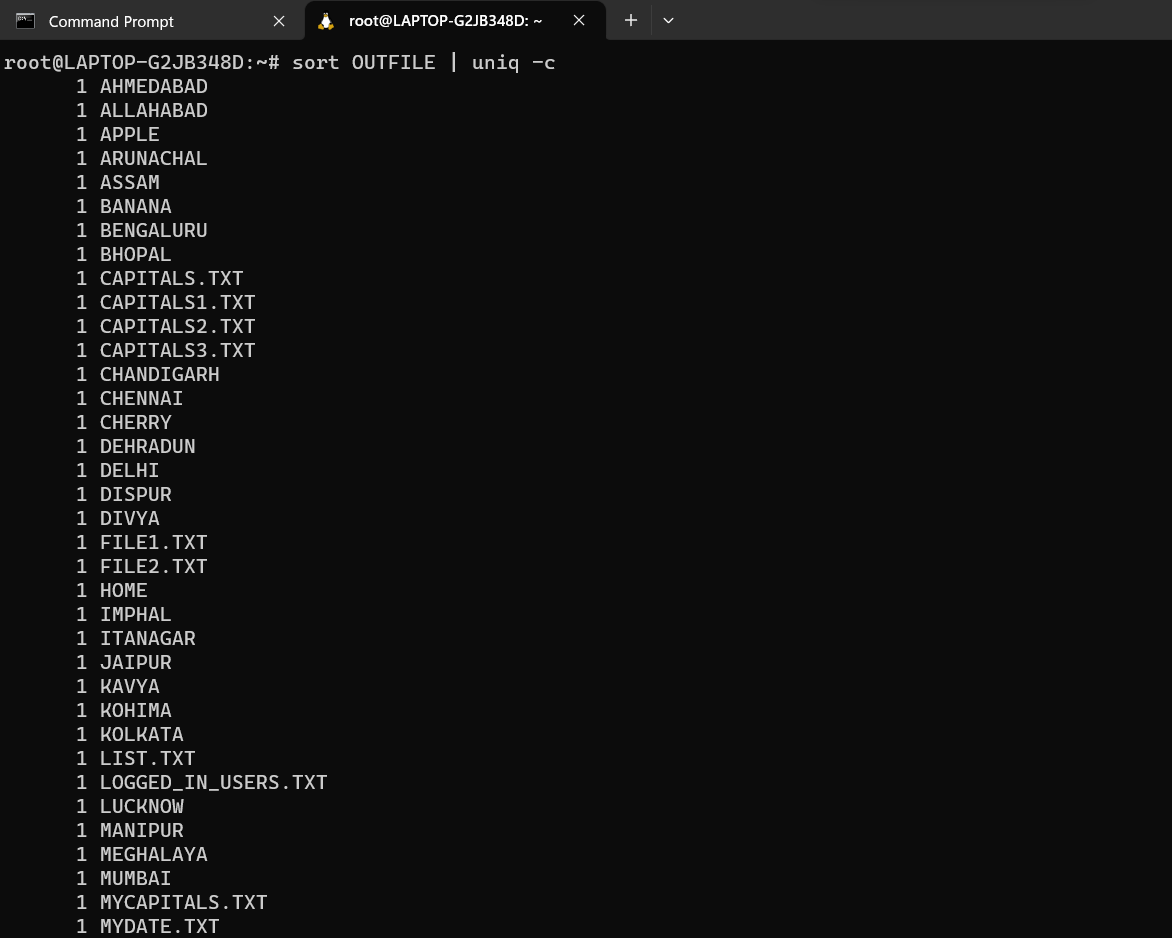
Problem Statement 6 )List all hidden files in the current directory



Problem Statement 7) Convert lower to upper case of given file.



Problem Statement 8) Display how many times lines are repeated in given file.



SHELL

Problem Statement1) Check whether a number is positive or negative

CODE:

#!/bin/bash

echo "enter a number"

read num

if [ $num -gt 0 ]

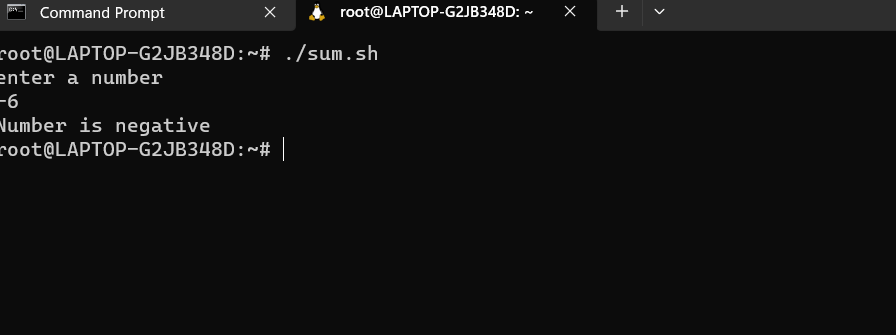
then

echo "number is positive"

else

echo "Number is negative"

fi



Problem Statement 2) Write arithmetic calculation program by using case structure.

#!/bin/bash

echo "Enter operator + -/ \* "

read op

echo "Enter 2 numbers"

read a b

case $op in

    +) result=$(($a+$b))

       ;;

    -)result=$(($a-$b))

      ;;

    \*)result=$(($a \* $b))

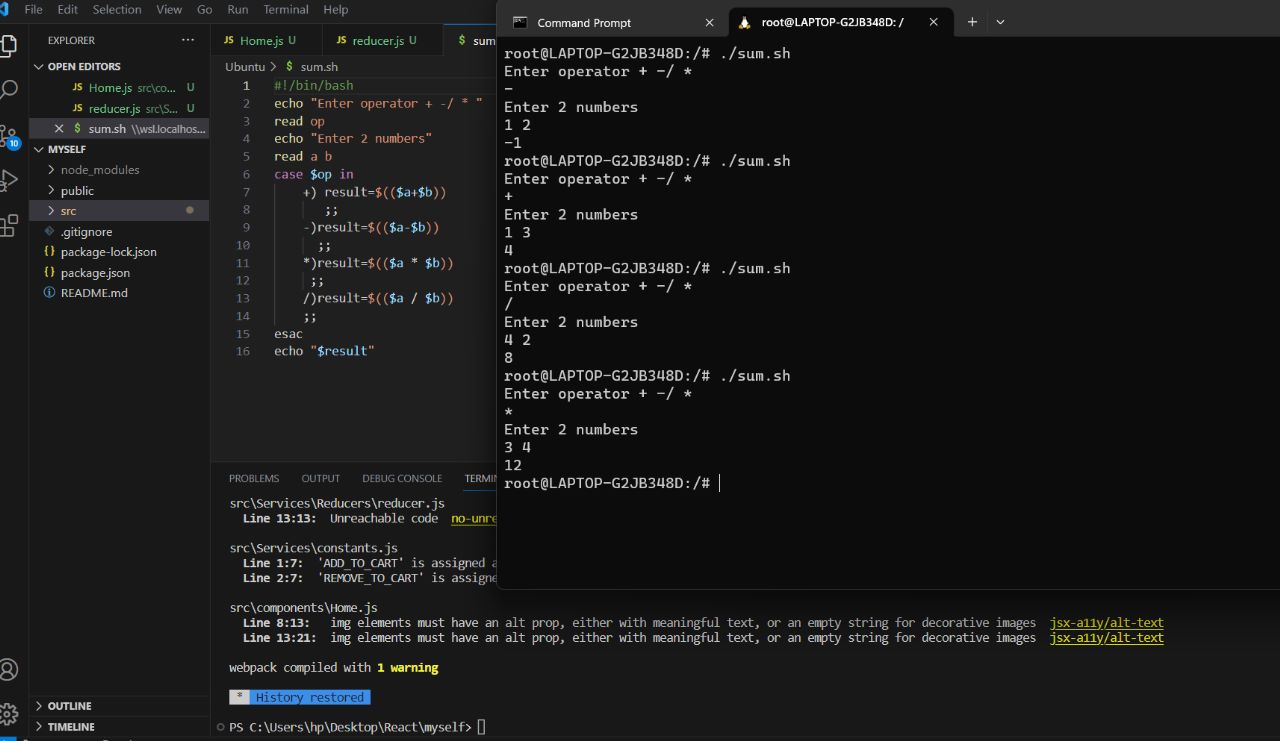
     ;;

    /)result=$(($a / $b))

    ;;

esac

echo "$result"



Problem Statement 3) Write a program to calc factorial of given no.

#!/bin/bash

echo "enter a number"

read num

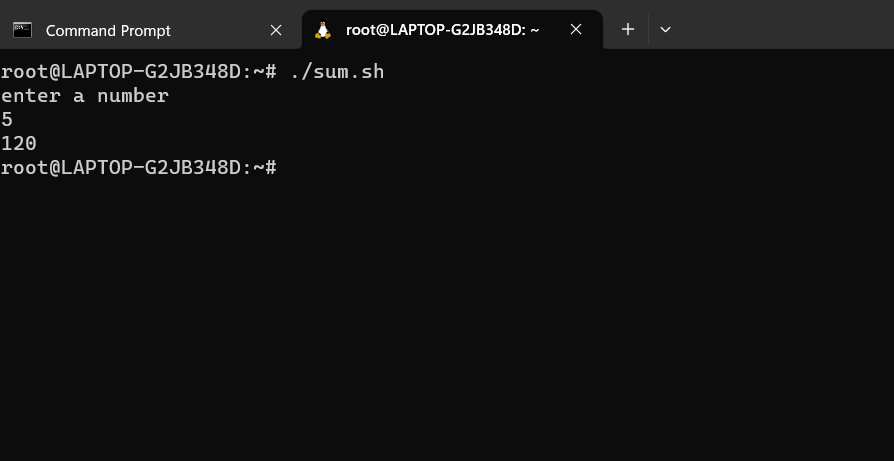
let fact=1;

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

    fact=$((fact \* i))

  done

echo "$fact"



Problem Statement 4) Check whether a string is palindrome or not

#!/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 ]

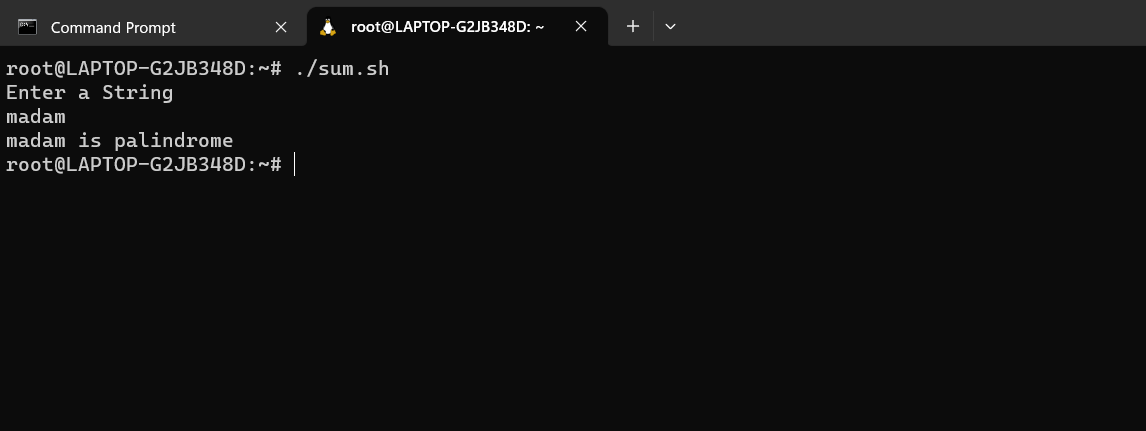
then

    echo "$input is palindrome"

else

    echo "$input is not palindrome"

fi



Problem Statement 5) Check whether a number is prime or not

#!/bin/bash

echo -e "Enter Number : \c"

read n

for((i=2; i<=$n/2; i++))

do

  ans=$(( n%i ))

  if [ $ans -eq 0 ]

  then

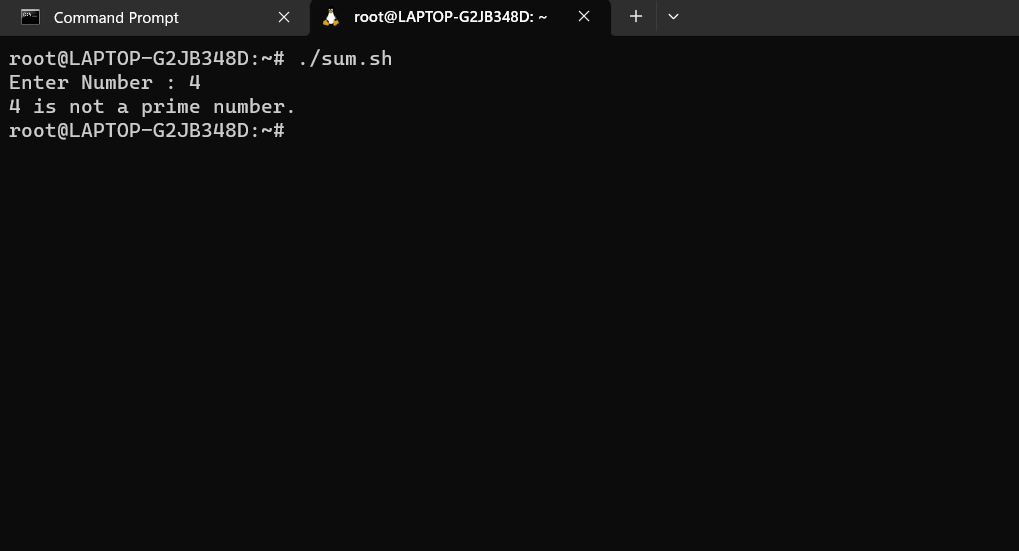
    echo "$n is not a prime number."

    exit 0

  fi

done

echo "$n is a prime number."



Problem Statement 6) To print reverse of given number.

echo enter number

read num1

num=0

while [ $num1 -gt 0 ]

do

num=$(expr $num \\* 10)

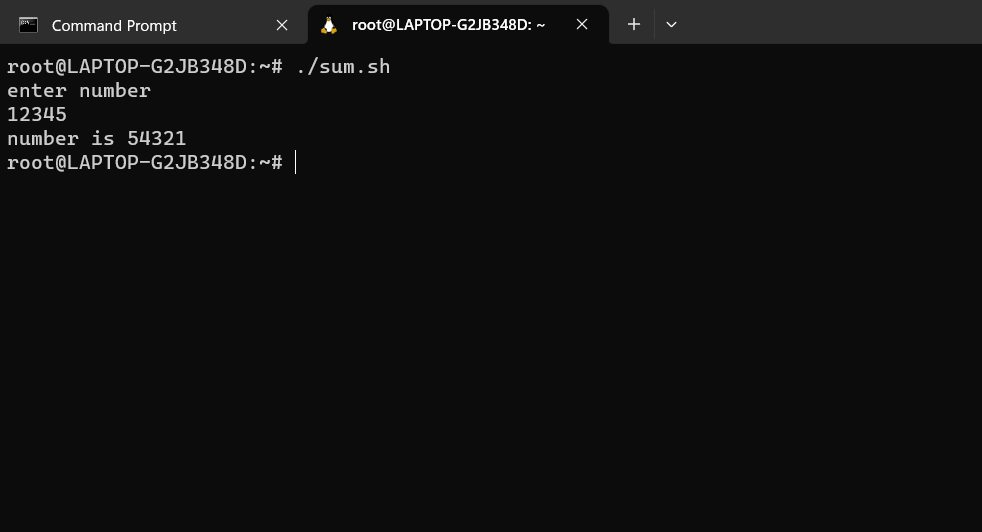
div=$(expr $num1 % 10)

num=$(expr $num + $div)

num1=$(expr $num1 / 10)

done

echo number is $num



Problem Statement 7) To check and count occurrence of a substring in a string using Command Line Arrgeument

#!/bin/bash

count\_substring\_occurrences() {

    local string="$1"

    local substring="$2"

    occurrences=$(( ( ${#string} - ${#substring} ) / ${#substring} ))

    echo $occurrences

}

if [ $# -eq 2 ]; then

    input\_string="$1"

    search\_substring="$2"

else

    echo "Enter a string:"

    read input\_string

    echo "Enter a substring:"

    read search\_substring

fi

result=$(count\_substring\_occurrences "$input\_string" "$search\_substring")

echo "Occurrences of '$search\_substring' in '$input\_string': $result"

