DTS Unix-Shell Scripting Capstone Assignment

Name: Atharva Athanikar 72287317E

Batch: DTS Batch-3

1) Write a shell script that folds long lines into 40 columns. Thus, any line that

exceeds 40 characters must be broken after 40th, a “\” is to be appended as the

indication of folding and the processing is to be continued with the residue. The

input is to be supplied through a text file created by the user.

Code:

#Program 1

input\_file=$1

while IFS= read -r line; do

while [ ${#line} -gt 40 ]; do

echo "${line:0:40}\\"

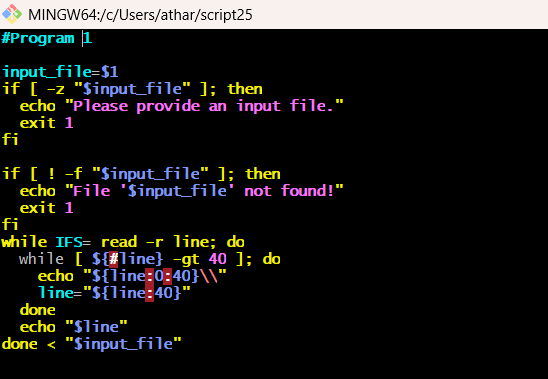
line="${line:40}"

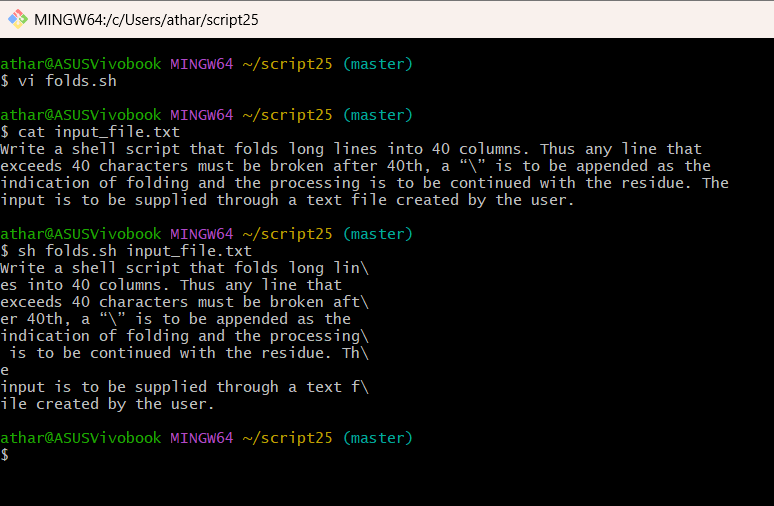
done

echo "$line"

done < "$input\_file"

OUTPUT:





2) Write a shell script to find a file/s that matches a pattern given as command

line argument in the home directory, display the contents of the file and copy the

file into the directory ~/mydir.

Code:

#Program 2

pattern=$1

home\_dir=~

destination=~/mydir

mkdir -p "$destination"

for file in "$home\_dir"/\*$pattern\*; do

if [ -f "$file" ]; then

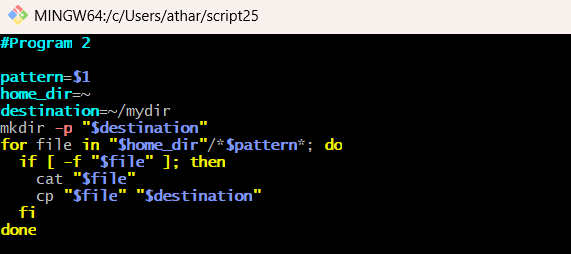
cat "$file"

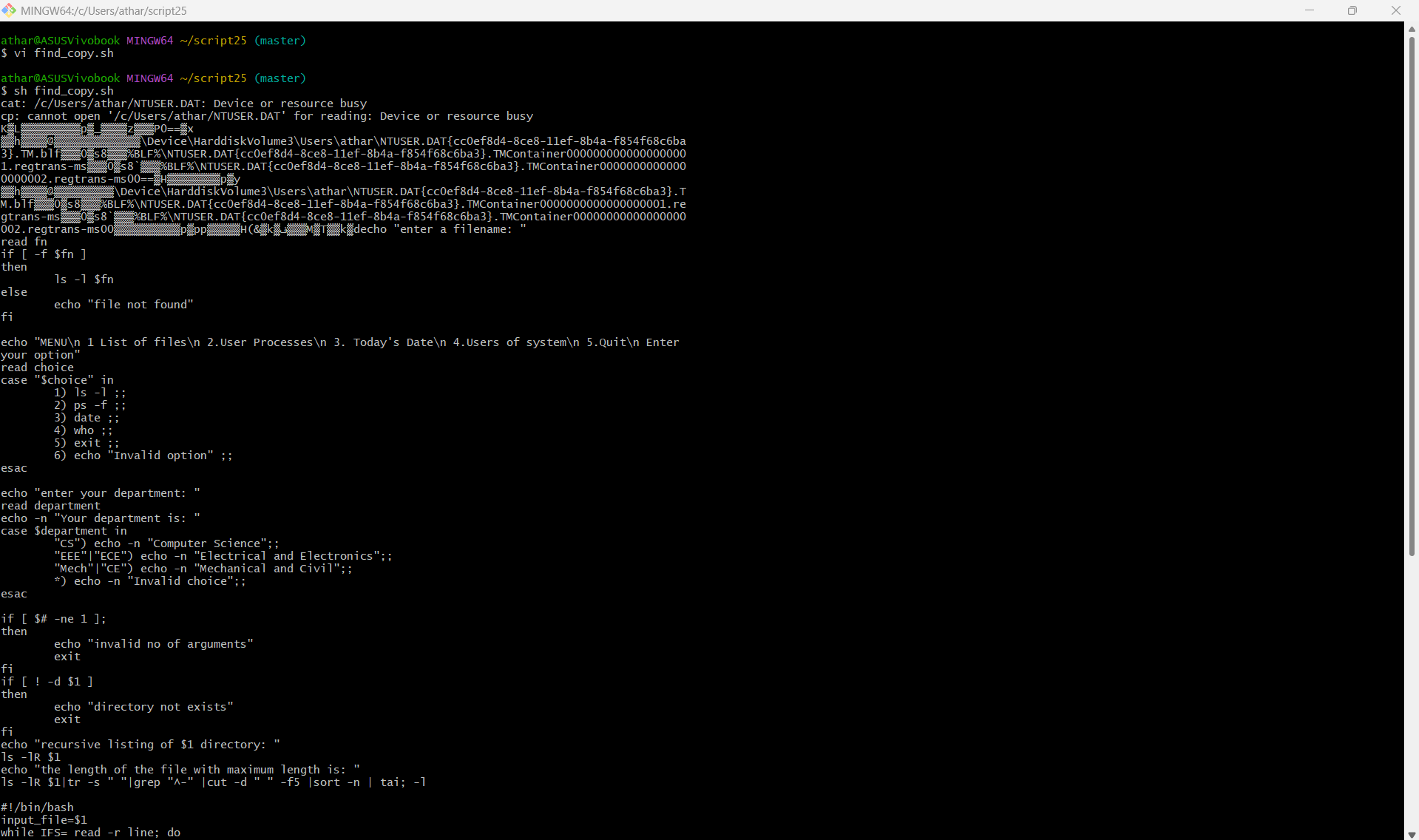
cp "$file" "$destination"

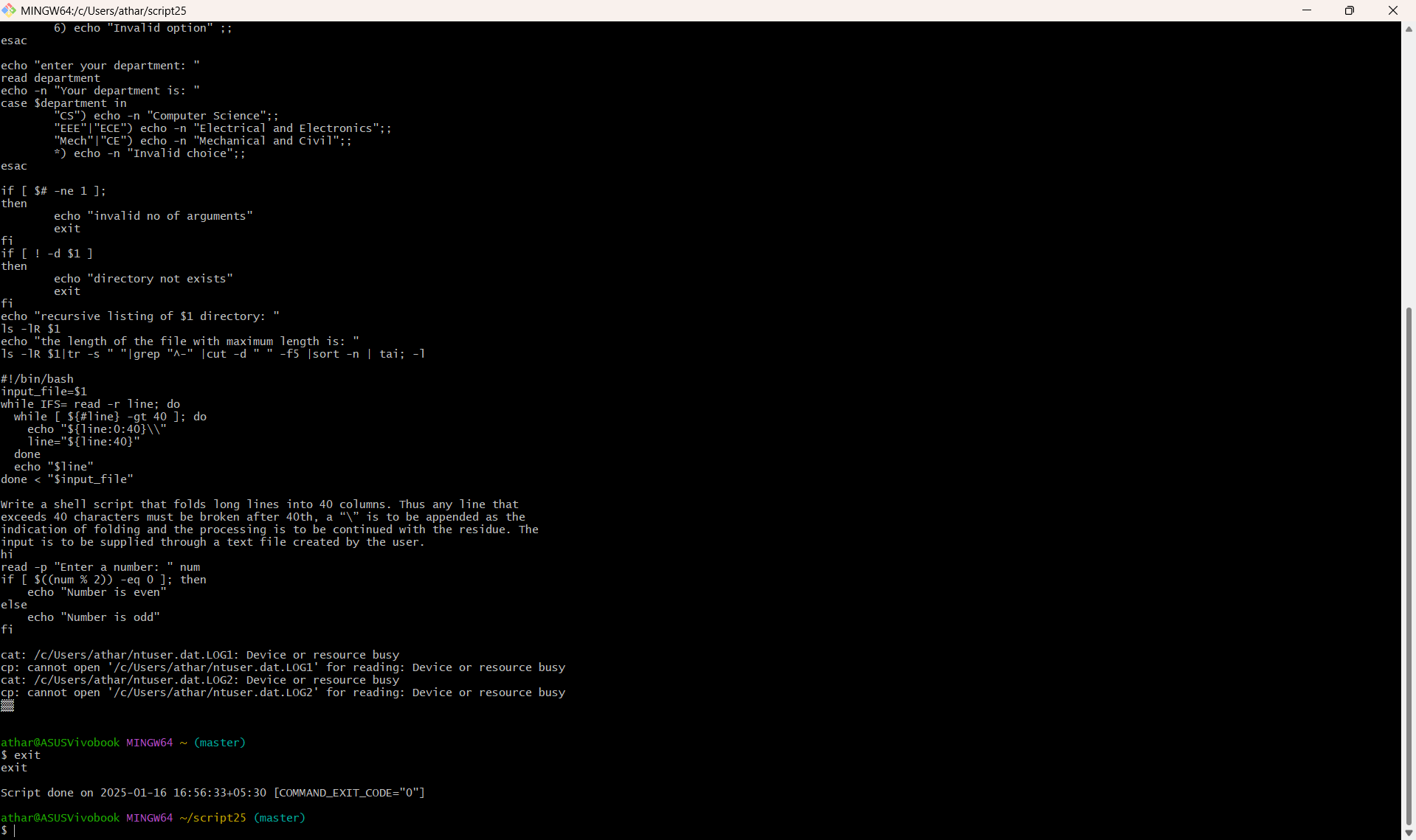
fi

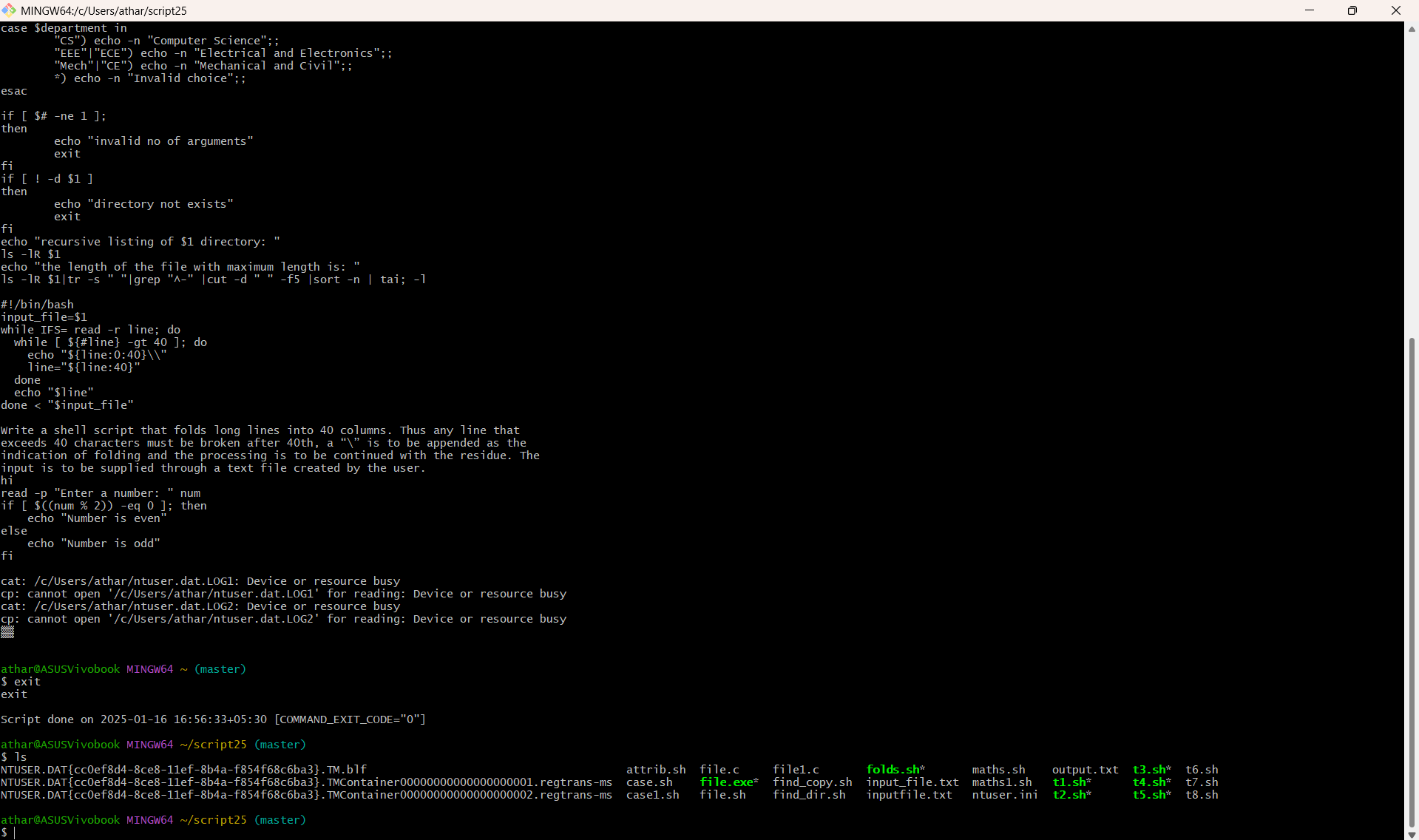
done

OUTPUT:









3) Write a shell script to compute GCD & LCM of two numbers.

Code:

#Program 3

gcd() {

while [ $2 -ne 0 ]; do

t=$2

b=$(( $1 % $2 ))

set -- $t $b

done

echo $1

}

lcm() {

echo $(( $1 \* $2 / $(gcd $1 $2) ))

}

read -p "Enter the first number: " num1

read -p "Enter the second number: " num2

if [[ ! $num1 =~ ^[0-9]+$ || ! $num2 =~ ^[0-9]+$ ]]; then

echo "Invalid input. Please enter positive integers."

exit 1

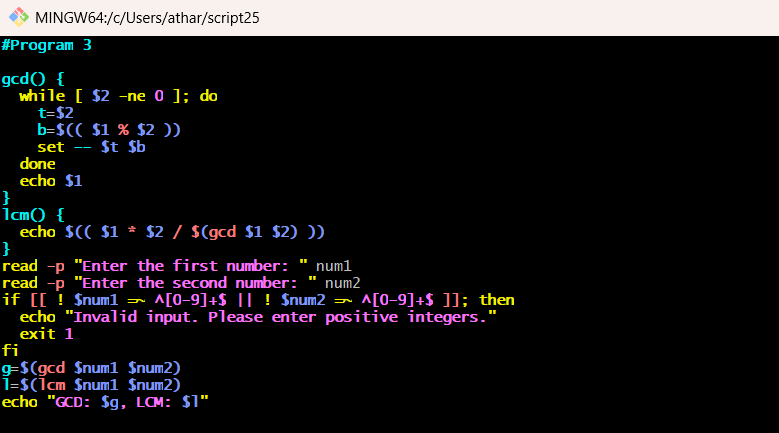
fi

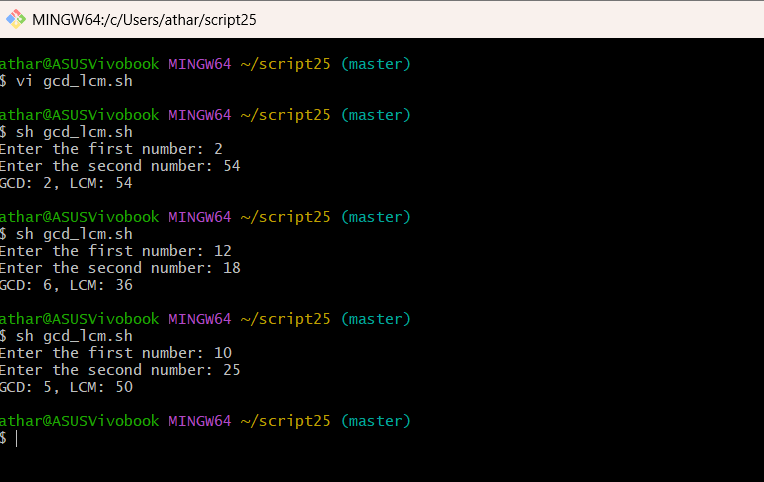
g=$(gcd $num1 $num2)

l=$(lcm $num1 $num2)

echo "GCD: $g, LCM: $l"

OUTPUT:





4) Write a shell script that accepts a pathname and creates all the components in

that pathname as directories. for example, if the script it named mpc,then the

command mpc a/b/c/d should create directories a,a/b,a/b/c,a/b/c/d.

Code:

#Program 4

if [ $# -eq 0 ]; then

echo "Usage: $0 pathname"

exit 1

fi

mkdir -p "$1"

OUTPUT:





5) Write a shell script to find whether a given number is prime.

Code:

#Program 5

read -p "Enter a number: " n

if ! [[ $n =~ ^[0-9]+$ && $n -gt 1 ]]; then

echo "Invalid input. Please enter a positive integer greater than 1."

exit 1

fi

for ((i=2; i\*i<=n; i++)); do

if ((n % i == 0)); then

echo "$n is not a prime number."

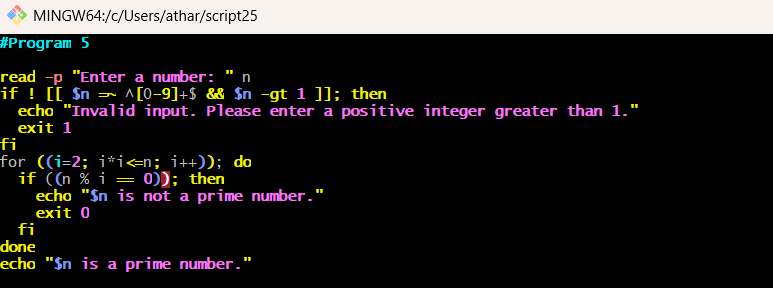
exit 0

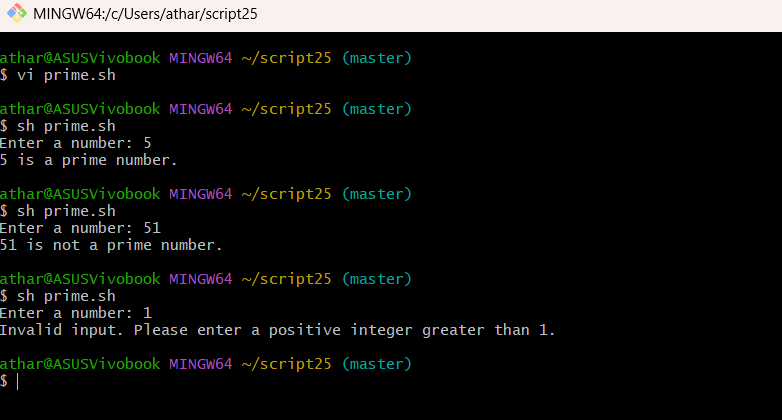
fi

done

echo "$n is a prime number."

OUTPUT:





6) Write a shell script that get executed display the message either “good

morning” or “good afternoon” or “good evening” depending upon time at which

the user logs-in.

Code:

#Program 6

hour=$(date +%H)

if [ $hour -lt 12 ]; then

echo "Good morning"

elif [ $hour -lt 18 ]; then

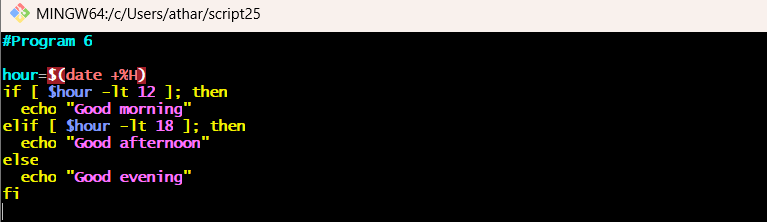
echo "Good afternoon"

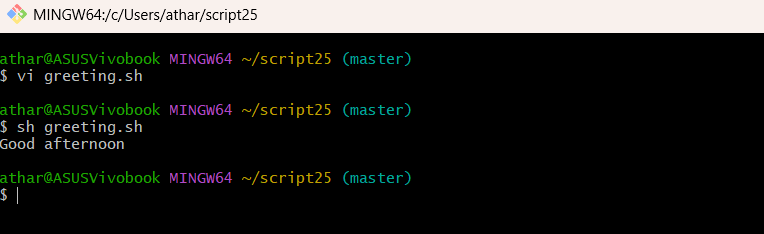
else

echo "Good evening"

fi

OUTPUT:





7) Write a shell script that accepts two file names as arguments, check if the

permissions for these files are identical and if the permissions are identical,

output common permissions and otherwise output each file names followed by

it’s permissions.

Code:

# Program 7

if [ $# -ne 2 ]; then

echo "Usage: $0 file1 file2"

exit 1

fi

perm1=$(stat -c %A "$1")

perm2=$(stat -c %A "$2")

if [ "$perm1" = "$perm2" ]; then

echo "Identical permissions: $perm1"

else

echo "$1: $perm1"

echo "$2: $perm2"

fi

OUTPUT:

