- echo "Hello, World!" -: it will print on shell
- name="Productive": -in name variable it will store Productive
- touch file.txt= it will create a file name with file.txt
- ls -a:- show also hidden file
- rm file.txt:- remove the file whose name file.txt
- cp file1.txt file2.txt:-copy the content of file1.txt in file2.txt.if file2.txt is not there it make
- mv file.txt /path/to/directory/
- chmod 755 script.sh:- change mode of owner-read,write,exute,Group;-read,excute, Other:-read,excute,
- grep "pattern" file.txt:-The **grep** utility searches the given input *files* selecting lines which match one or more *patterns*. The type of patterns is controlled by the options specified. By default, a pattern matches an input line if any regular expression (RE) in the pattern matches the input line without its trailing newline. A null RE matches every line. Each input line that matches at least one of the patterns is written to the standard output.
- kill PID:- to kill themprocess in linux
- mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt:- driectory is created with name mydir and went to directory mydir a file is created with file name file.txt and on shell printed Hello world
- ls -l | grep ".txt"
- cat file1.txt file2.txt | sort | uniq
- ls -l | grep "^d": shows only directory permission
- grep -r "pattern" /path/to/directory/
- cat file1.txt file2.txt | sort | uniq –d
- chmod 644 file.txt
- cp -r source_directory destination_directory
- find /path/to/search -name "*.txt"
- chmod u+x file.txt:- give permission to user mode to exute
- echo $PATH: it gives all details about path

$PATH is a environment variable that is file location-related. When one types a command to run, the system looks for it in the directories specified by PATH in the order specified. You can view the directories specified by typing echo $PATH in the terminal.

Identify True or False:

1. ls is used to list files and directories in a directory.false

   ls used lists files and directories

2. mv is used to move files and directories. true

You can use the mv command to move files within the same file system or between file systems

2. cd is used to copy files and directories.: false
   the cd command is used to change directories, not to copy files and directories. The cp command is used to copy files and directories.
3. pwd stands for "print working directory" and displays the current directory. True
   That's correct, "pwd" stands for "print working directory" and is a command used in Unix-like operating systems to display the current directory you are in on the file system;

5. grep is used to search for patterns in files.True

6. chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.:True

7. mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist.

8. rm -rf file.txt deletes a file forcefully without confirmation.++++yes

Identify the Incorrect Commands:

1. chmodx is used to change file permissions. Incorrect correct is chmod

2. cpy is used to copy files and directories. Incorrect correct is cp

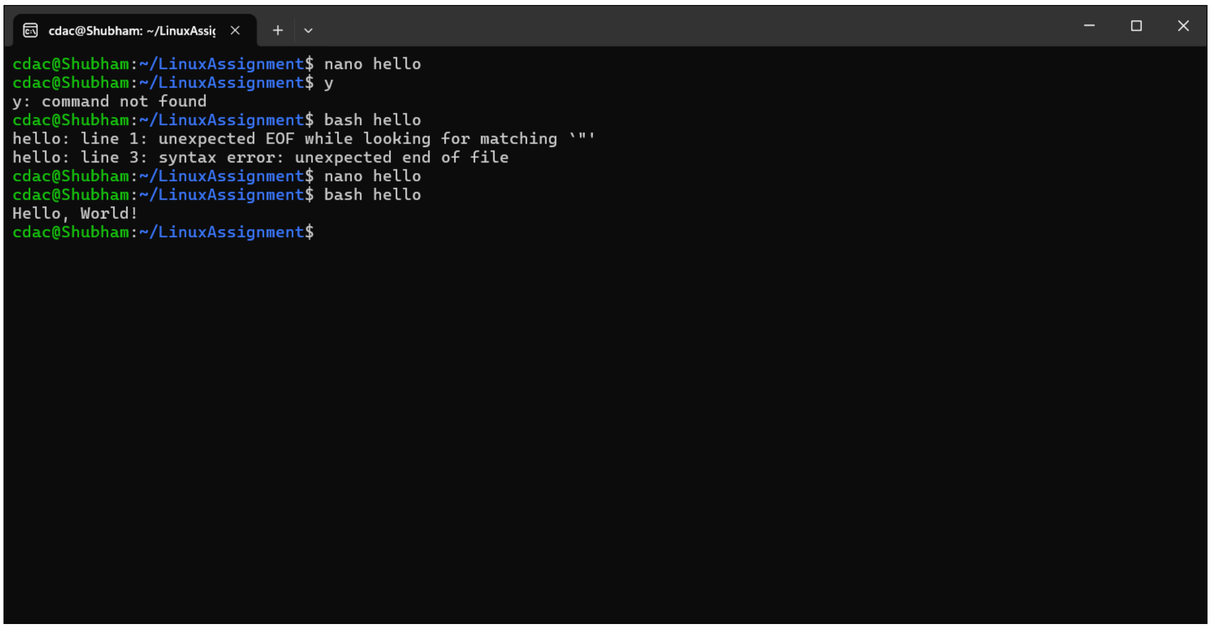
3. mkfile is used to create a new file. Incorrect correct is mkdir

4. catx is used to concatenate files. Incorrect correct is cat

5. rn is used to rename files. Incorrect correct is rm

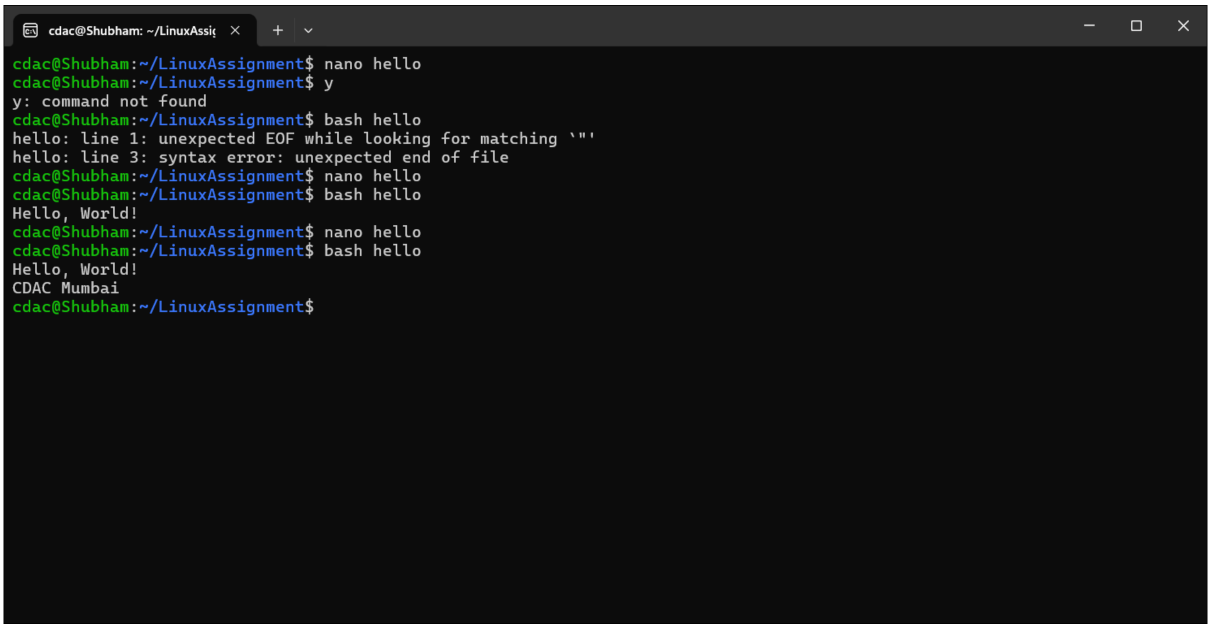
Part c

Question 1: Write a shell script that prints "Hello, World!" to the terminal.



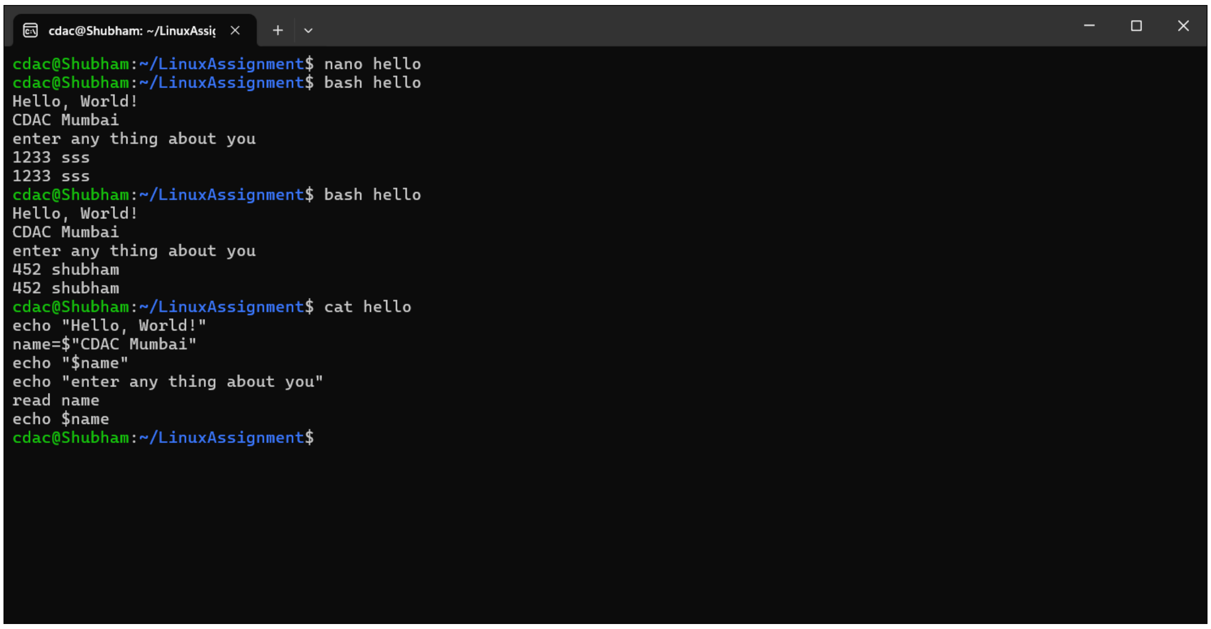
```
cdac@Shubham:~/LinuxAssignment$ nano hello
cdac@Shubham:~/LinuxAssignment$ y
y: command not found
cdac@Shubham:~/LinuxAssignment$ bash hello
hello: line 1: unexpected EOF while looking for matching `"'
hello: line 3: syntax error: unexpected end of file
cdac@Shubham:~/LinuxAssignment$ nano hello
cdac@Shubham:~/LinuxAssignment$ bash hello
Hello, World!
cdac@Shubham:~/LinuxAssignment$
```

Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.



```
cdac@Shubham:~/LinuxAssignment$ nano hello
cdac@Shubham:~/LinuxAssignment$ y
y: command not found
cdac@Shubham:~/LinuxAssignment$ bash hello
hello: line 1: unexpected EOF while looking for matching `"'
hello: line 3: syntax error: unexpected end of file
cdac@Shubham:~/LinuxAssignment$ nano hello
cdac@Shubham:~/LinuxAssignment$ bash hello
Hello, World!
cdac@Shubham:~/LinuxAssignment$ nano hello
cdac@Shubham:~/LinuxAssignment$ bash hello
Hello, World!
CDAC Mumbai
cdac@Shubham:~/LinuxAssignment$
```

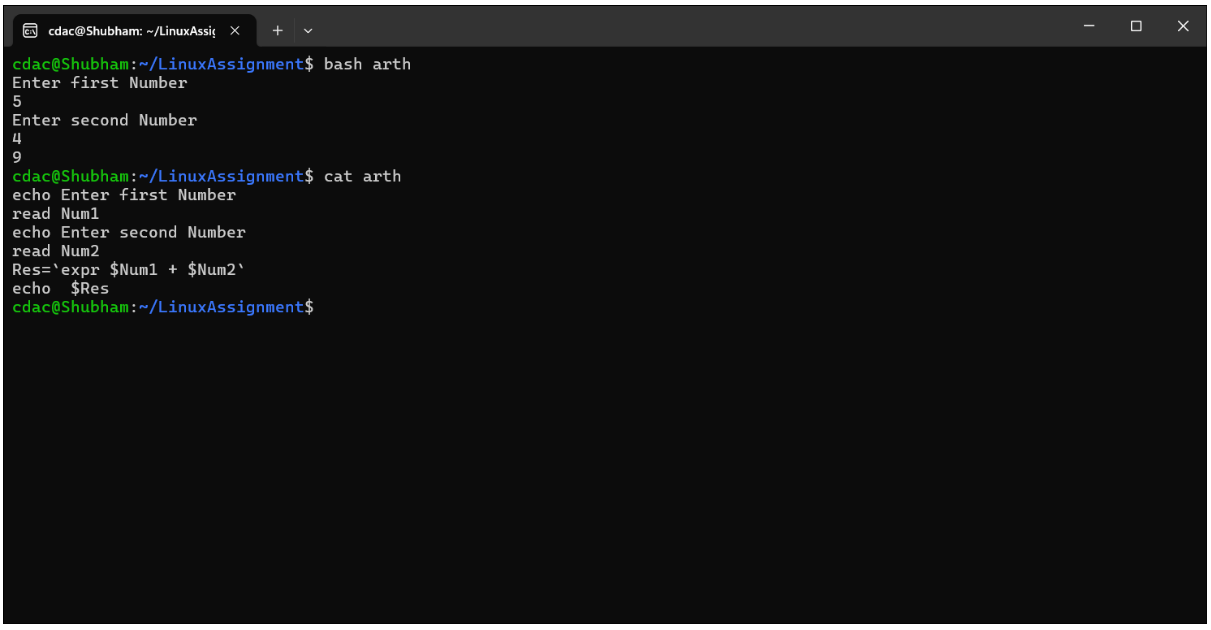
Question 3: Write a shell script that takes a number as input from the user and prints it.



```
cdac@Shubham:~/LinuxAssignment$ nano hello
cdac@Shubham:~/LinuxAssignment$ bash hello
Hello, World!
CDAC Mumbai
enter any thing about you
1233 sss
1233 sss
cdac@Shubham:~/LinuxAssignment$ bash hello
Hello, World!
CDAC Mumbai
enter any thing about you
452 shubham
452 shubham
cdac@Shubham:~/LinuxAssignment$ cat hello
echo "Hello, World!"
name=$"CDAC Mumbai"
echo "$name"
echo "enter any thing about you"
read name
echo $name
cdac@Shubham:~/LinuxAssignment$
```

Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the

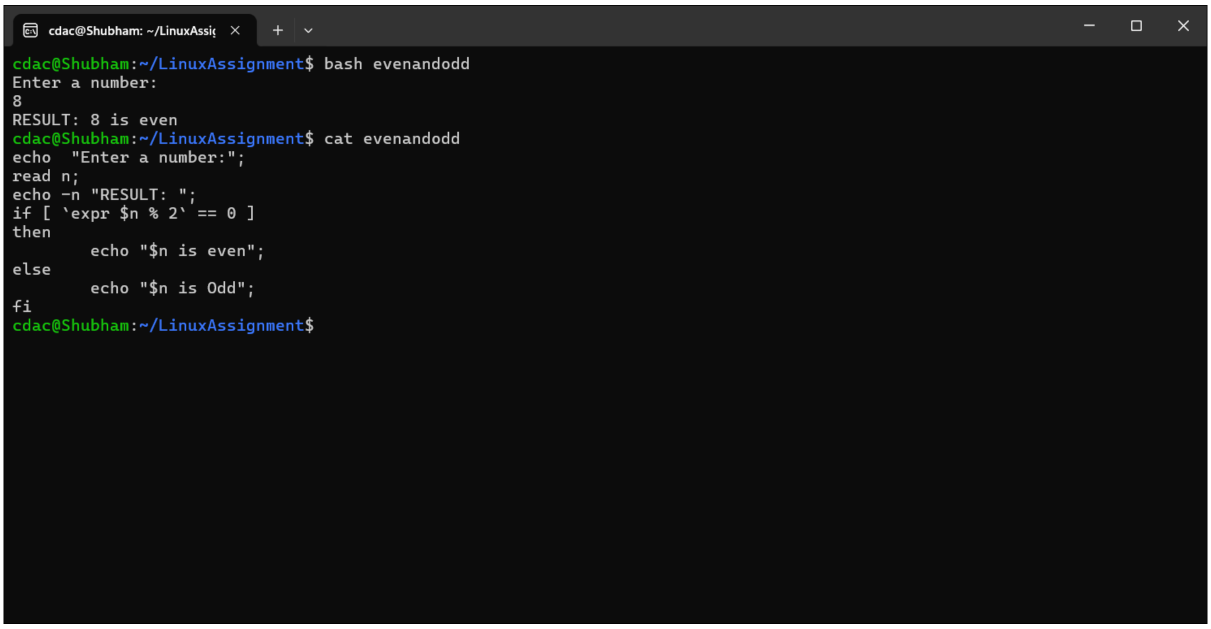
result.



```
cdac@Shubham:~/LinuxAssignment$ bash arth
Enter first Number
5
Enter second Number
4
9
cdac@Shubham:~/LinuxAssignment$ cat arth
echo Enter first Number
read Num1
echo Enter second Number
read Num2
Res=`expr $Num1 + $Num2`
echo  $Res
cdac@Shubham:~/LinuxAssignment$
```

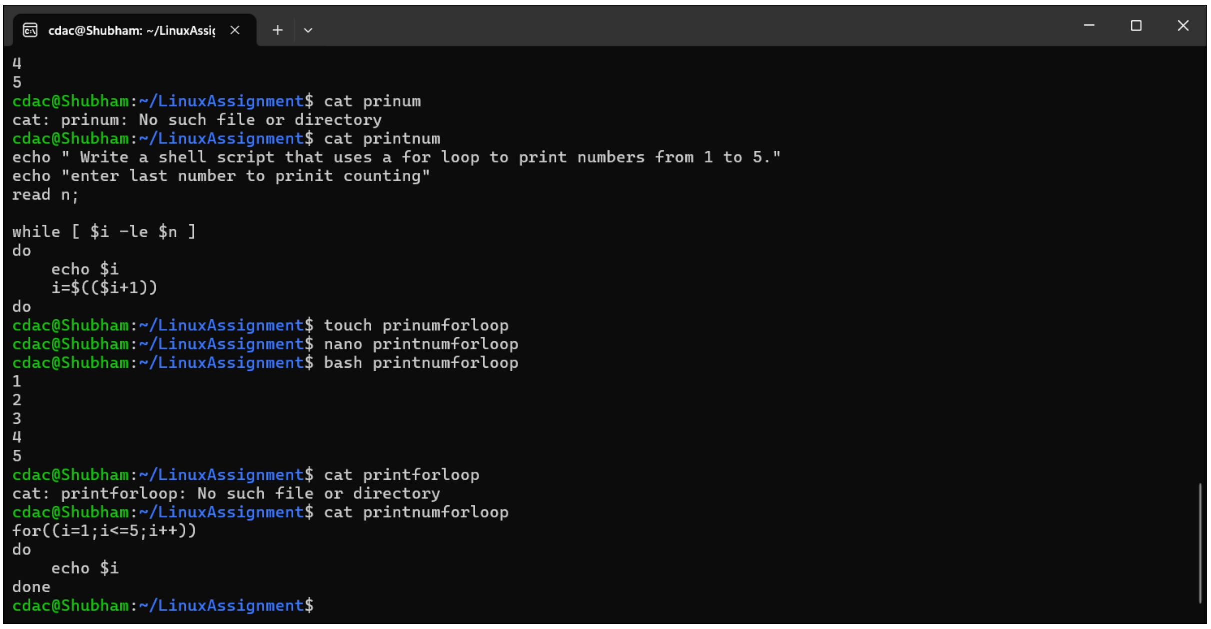
Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise

prints "Odd".



```
cdac@Shubham:~/LinuxAssignment$ bash evenandodd
Enter a number:
8
RESULT: 8 is even
cdac@Shubham:~/LinuxAssignment$ cat evenandodd
echo  "Enter a number:";
read n;
echo -n "RESULT: ";
if [ `expr $n % 2` == 0 ]
then
        echo "$n is even";
else
        echo "$n is Odd";
fi
cdac@Shubham:~/LinuxAssignment$
```

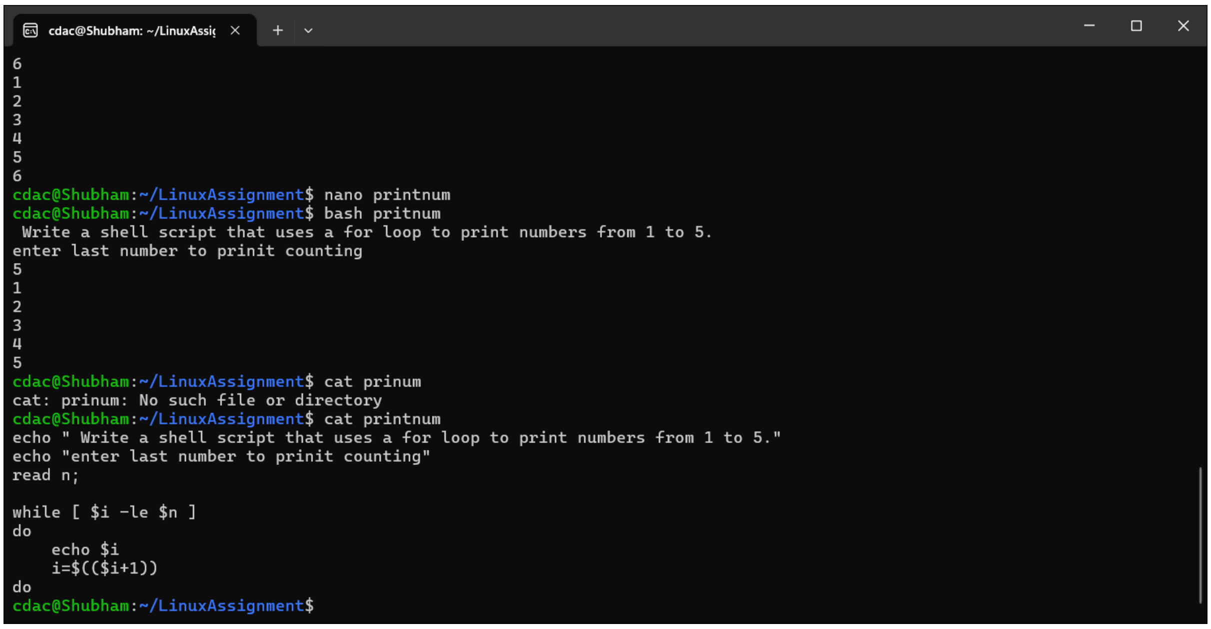
Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.



```
4
5
cdac@Shubham:~/LinuxAssignment$ cat prinum
cat: prinum: No such file or directory
cdac@Shubham:~/LinuxAssignment$ cat printnum
echo " Write a shell script that uses a for loop to print numbers from 1 to 5."
echo "enter last number to prinit counting"
read n;

while [ $i -le $n ]
do
    echo $i
    i=$(($i+1))
do
cdac@Shubham:~/LinuxAssignment$ touch prinumforloop
cdac@Shubham:~/LinuxAssignment$ nano printnumforloop
cdac@Shubham:~/LinuxAssignment$ bash printnumforloop
1
2
3
4
5
cdac@Shubham:~/LinuxAssignment$ cat printforloop
cat: printforloop: No such file or directory
cdac@Shubham:~/LinuxAssignment$ cat printnumforloop
for((i=1;i<=5;i++))
do
    echo $i
done
cdac@Shubham:~/LinuxAssignment$
```

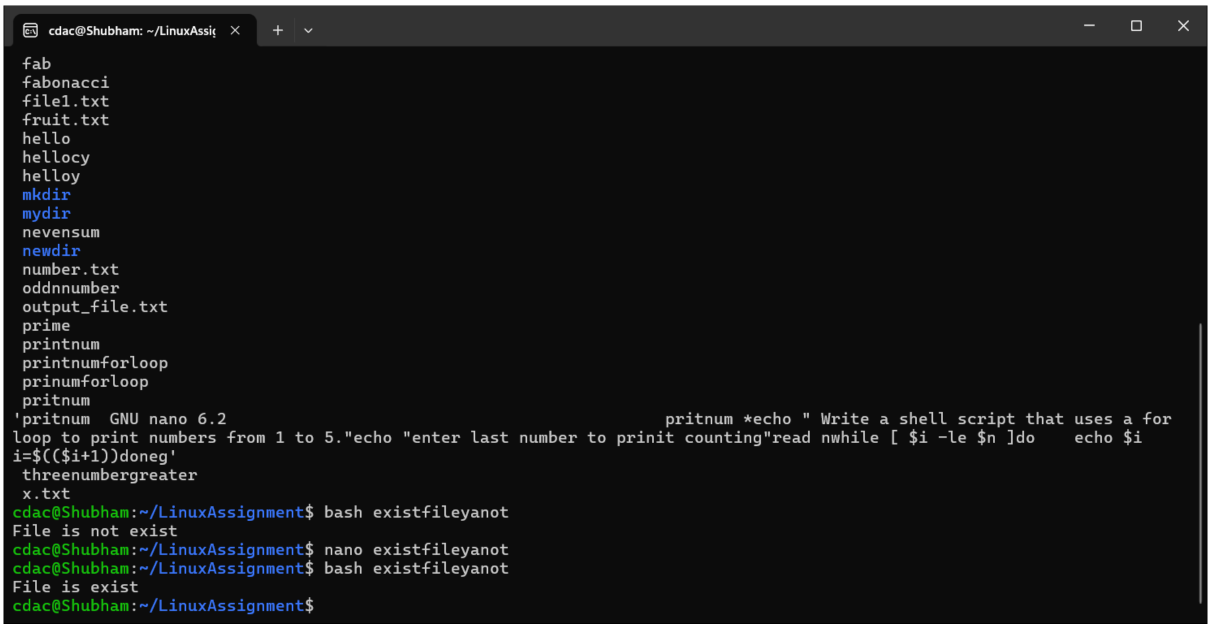
Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.



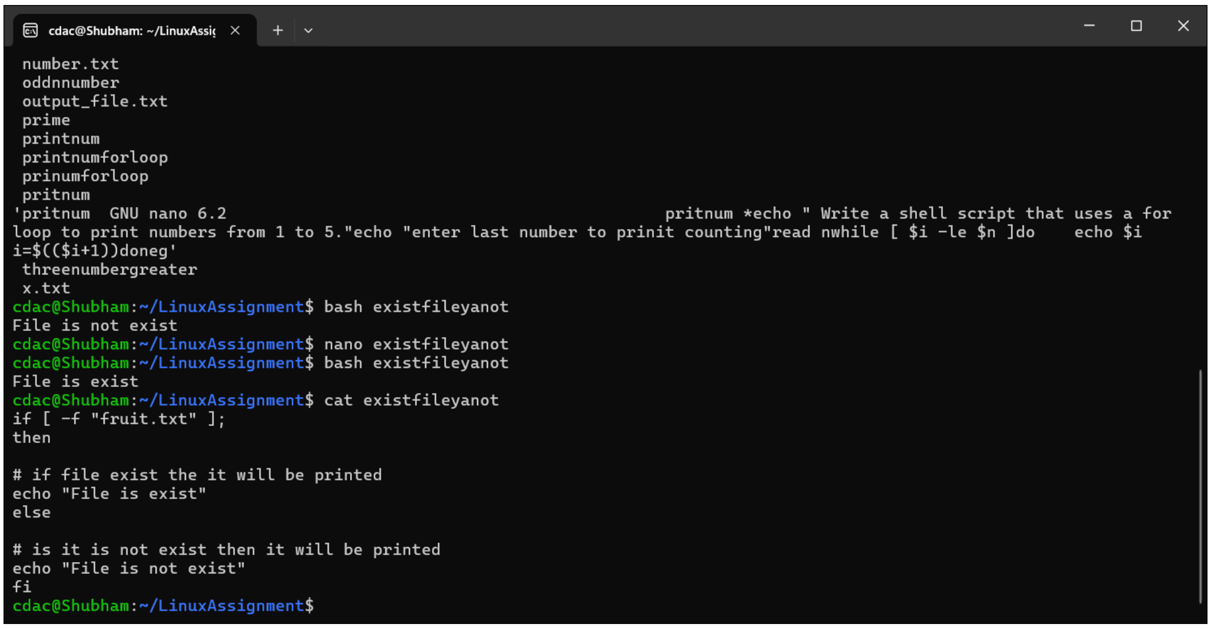
```
6
1
2
3
4
5
6
cdac@Shubham:~/LinuxAssignment$ nano printnum
cdac@Shubham:~/LinuxAssignment$ bash pritnum
 Write a shell script that uses a for loop to print numbers from 1 to 5.
enter last number to prinit counting
5
1
2
3
4
5
cdac@Shubham:~/LinuxAssignment$ cat prinum
cat: prinum: No such file or directory
cdac@Shubham:~/LinuxAssignment$ cat printnum
echo " Write a shell script that uses a for loop to print numbers from 1 to 5."
echo "enter last number to prinit counting"
read n;

while [ $i -le $n ]
do
    echo $i
    i=$(($i+1))
do
cdac@Shubham:~/LinuxAssignment$
```

Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".



```
 fab
 fabonacci
 file1.txt
 fruit.txt
 hello
 hellocy
 helloy
 mkdir
 mydir
 nevensum
 newdir
 number.txt
 oddnnumber
 output_file.txt
 prime
 printnum
 printnumforloop
 prinumforloop
 pritnum
'pritnum  GNU nano 6.2                                          pritnum *echo " Write a shell script that uses a for
loop to print numbers from 1 to 5."echo "enter last number to prinit counting"read nwhile [ $i -le $n ]do    echo $i
i=$(($i+1))doneg'
 threenumbergreater
 x.txt
cdac@Shubham:~/LinuxAssignment$ bash existfileyanot
File is not exist
cdac@Shubham:~/LinuxAssignment$ nano existfileyanot
cdac@Shubham:~/LinuxAssignment$ bash existfileyanot
File is exist
cdac@Shubham:~/LinuxAssignment$
```



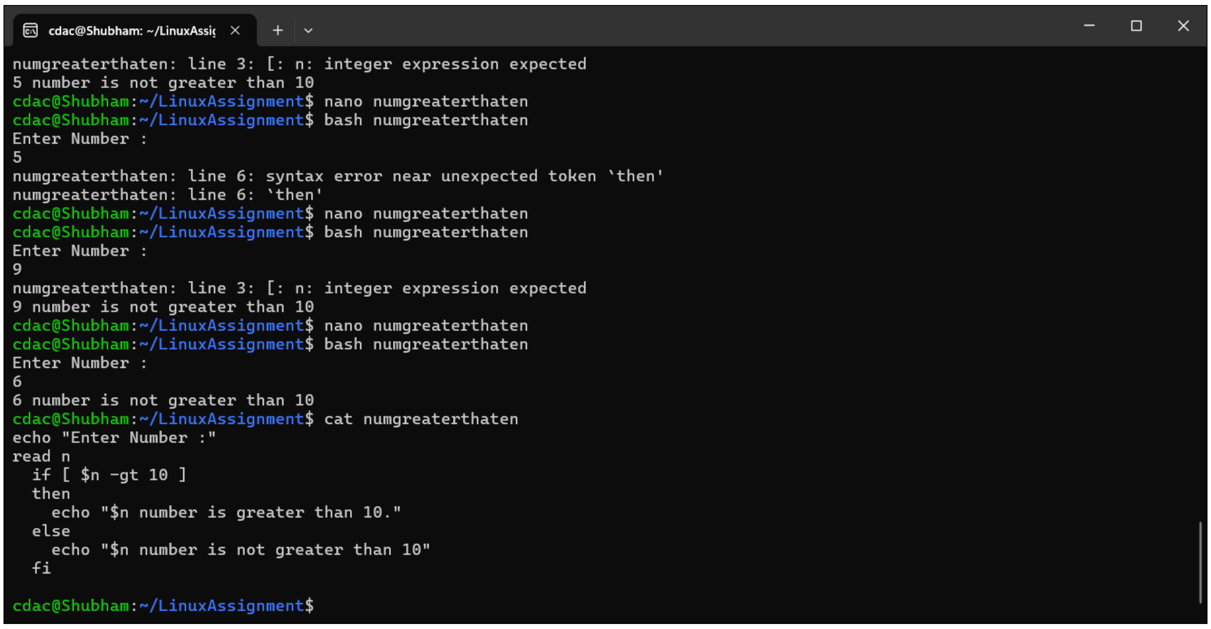
```
 number.txt
 oddnnumber
 output_file.txt
 prime
 printnum
 printnumforloop
 prinumforloop
 pritnum
'pritnum  GNU nano 6.2                                              pritnum *echo " Write a shell script that uses a for
loop to print numbers from 1 to 5."echo "enter last number to prinit counting"read nwhile [ $i -le $n ]do    echo $i
i=$(($i+1))doneg'
 threenumbergreater
 x.txt
cdac@Shubham:~/LinuxAssignment$ bash existfileyanot
File is not exist
cdac@Shubham:~/LinuxAssignment$ nano existfileyanot
cdac@Shubham:~/LinuxAssignment$ bash existfileyanot
File is exist
cdac@Shubham:~/LinuxAssignment$ cat existfileyanot
if [ -f "fruit.txt" ];
then

# if file exist the it will be printed
echo "File is exist"
else

# is it is not exist then it will be printed
echo "File is not exist"
fi
cdac@Shubham:~/LinuxAssignment$
```

Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and

prints a message accordingly.\

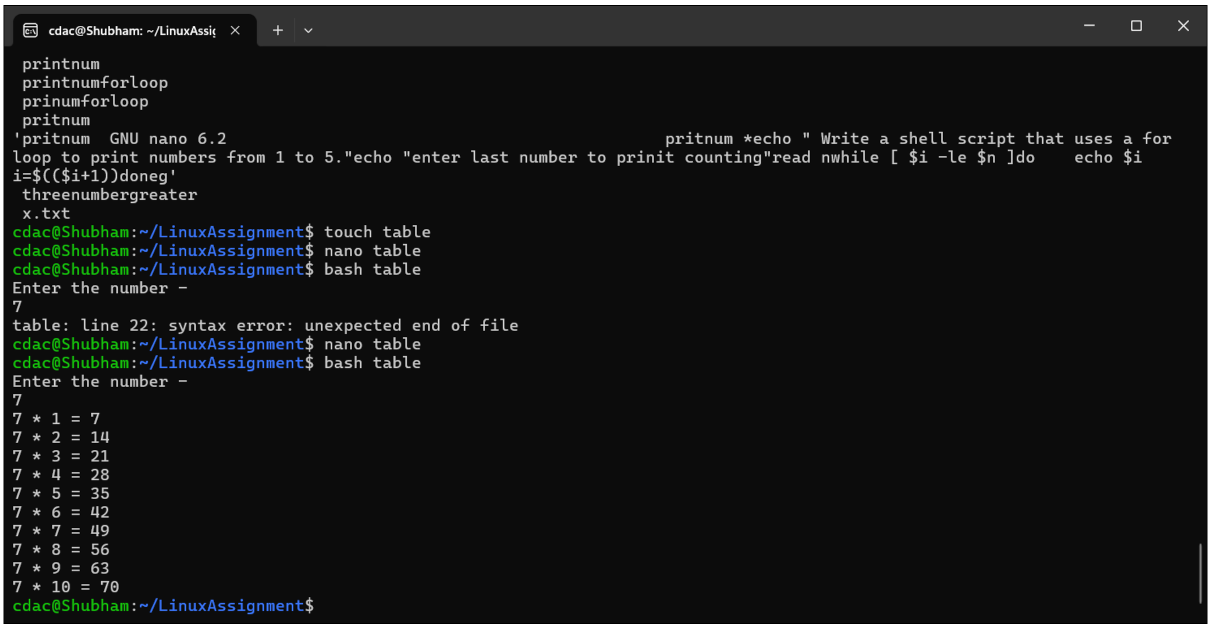


```
numgreaterthaten: line 3: [: n: integer expression expected
5 number is not greater than 10
cdac@Shubham:~/LinuxAssignment$ nano numgreaterthaten
cdac@Shubham:~/LinuxAssignment$ bash numgreaterthaten
Enter Number :
5
numgreaterthaten: line 6: syntax error near unexpected token `then'
numgreaterthaten: line 6: `then'
cdac@Shubham:~/LinuxAssignment$ nano numgreaterthaten
cdac@Shubham:~/LinuxAssignment$ bash numgreaterthaten
Enter Number :
9
numgreaterthaten: line 3: [: n: integer expression expected
9 number is not greater than 10
cdac@Shubham:~/LinuxAssignment$ nano numgreaterthaten
cdac@Shubham:~/LinuxAssignment$ bash numgreaterthaten
Enter Number :
6
6 number is not greater than 10
cdac@Shubham:~/LinuxAssignment$ cat numgreaterthaten
echo "Enter Number :"
read n
  if [ $n -gt 10 ]
  then
    echo "$n number is greater than 10."
  else
    echo "$n number is not greater than 10"
  fi

cdac@Shubham:~/LinuxAssignment$
```

Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers

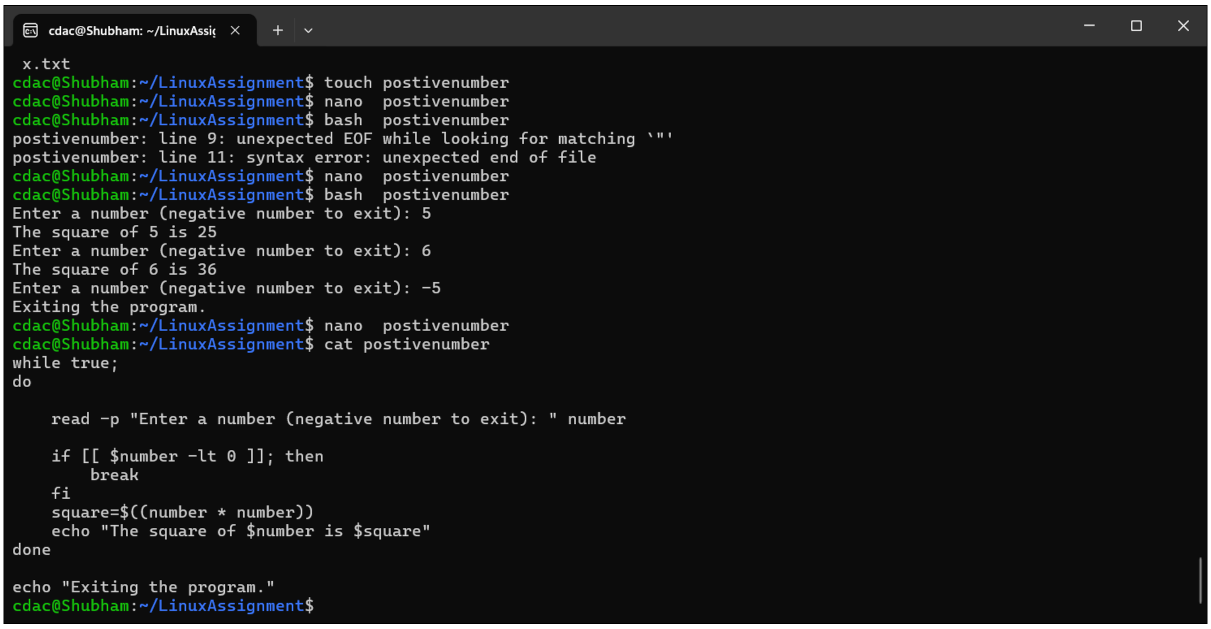
from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.


```
 printnum
 printnumforloop
 prinumforloop
 pritnum
'pritnum  GNU nano 6.2                                        pritnum *echo " Write a shell script that uses a for
loop to print numbers from 1 to 5."echo "enter last number to prinit counting"read nwhile [ $i -le $n ]do    echo $i
i=$(($i+1))doneg'
 threenumbergreater
 x.txt
cdac@Shubham:~/LinuxAssignment$ touch table
cdac@Shubham:~/LinuxAssignment$ nano table
cdac@Shubham:~/LinuxAssignment$ bash table
Enter the number -
7
table: line 22: syntax error: unexpected end of file
cdac@Shubham:~/LinuxAssignment$ nano table
cdac@Shubham:~/LinuxAssignment$ bash table
Enter the number -
7
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70
cdac@Shubham:~/LinuxAssignment$
```

Write a shell script that uses a while loop to read numbers from the user until the user enters

a negative number. For each positive number entered, print its square. Use the break statement to exit the

loop when a negative number is entered.


```
 x.txt
cdac@Shubham:~/LinuxAssignment$ touch postivenumber
cdac@Shubham:~/LinuxAssignment$ nano  postivenumber
cdac@Shubham:~/LinuxAssignment$ bash  postivenumber
postivenumber: line 9: unexpected EOF while looking for matching `"'
postivenumber: line 11: syntax error: unexpected end of file
cdac@Shubham:~/LinuxAssignment$ nano  postivenumber
cdac@Shubham:~/LinuxAssignment$ bash  postivenumber
Enter a number (negative number to exit): 5
The square of 5 is 25
Enter a number (negative number to exit): 6
The square of 6 is 36
Enter a number (negative number to exit): -5
Exiting the program.
cdac@Shubham:~/LinuxAssignment$ nano  postivenumber
cdac@Shubham:~/LinuxAssignment$ cat postivenumber
while true;
do

    read -p "Enter a number (negative number to exit): " number

    if [[ $number -lt 0 ]]; then
        break
    fi
    square=$((number * number))
    echo "The square of $number is $square"
done

echo "Exiting the program."
cdac@Shubham:~/LinuxAssignment$
```

Part E

Adobe Scan 02 Mar 2025.pdf

Steps

Navigate to your working directory or create a new folder by the name **shell_scripts** and navigate to it using the following command

```
mkdir shell_scripts && cd shell_scripts
```

Create a file by the name **if.sh**. Use the following command

```
touch if.sh
```

Make the file executable using the following command

```
chmod +x if.sh
```

Open the file with nano as shown below

```
nano if.sh
```

Enter the following line at the top of your script. It will allow you to execute this file using bash

```
#!/usr/bin/bash
```

Before we continue with our practical example, it is good to know some of the comparison operators that we will use in this example. We also need to know the syntax of an if statement so that we can use it.

Here are some of the comparison operators:

Integer Comparison

-eq : *equal to*

-ne : *not equal to*

-gt : *greater than*

-ge : *greater than or equal to*

-lt : *less than*

-le : *less or equal*

< : *less than*

> *: greater than*

<= : *Less than or equal to*

>= *: Greater than or equal*

String Comparison

= : *equal to*

== : *equal to*

!= : *not equal to*

< : *Less than*

> *: greater than*

-z : *string is null*

Below is the syntax for an if statement

```
if [[ condition ]]
then
  # Do something
elif [[ another_condition ]]
then
  # Do this instead
else
  # Do this if neither of the above is met
fi
```