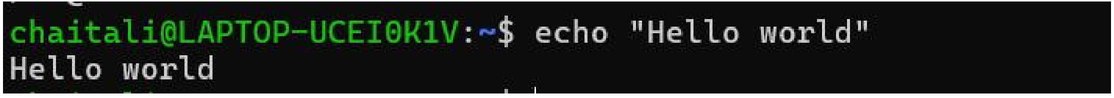
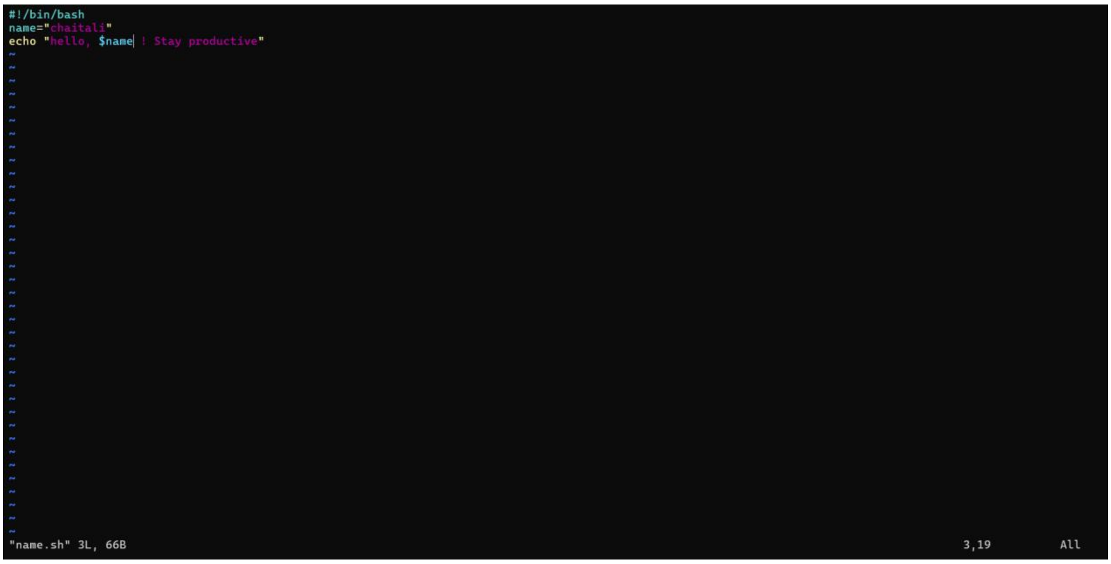
PART A

1.echo "Hello, World



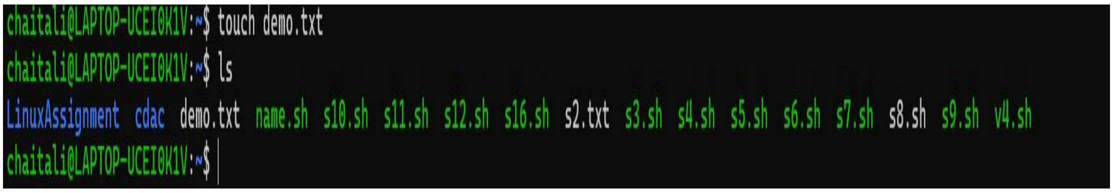
```
chaitali@LAPTOP-UCEI0K1V:~$ echo "Hello world"
Hello world
```

2.name="Productive"
-A shell with variable ‘name’ with value Productive is created



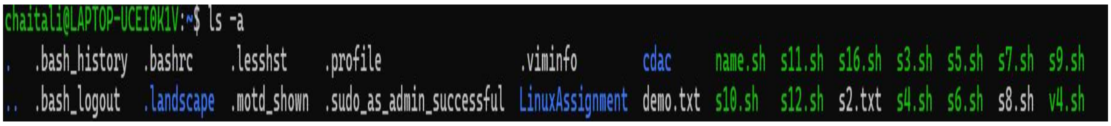
```
#!/bin/bash
name="chaitali"
echo "hello, $name ! Stay productive"
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"name.sh" 3L, 66B                                                   3,19          All
```

3.touch file.txt
-creates empty file named file.txt



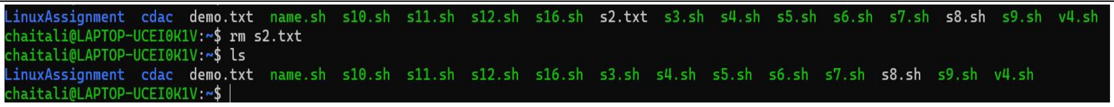
```
chaitali@LAPTOP-UCEI0K1V:~$ touch demo.txt
chaitali@LAPTOP-UCEI0K1V:~$ ls
LinuxAssignment  cdac  demo.txt  name.sh  s10.sh  s11.sh  s12.sh  s16.sh  s2.txt  s3.sh  s4.sh  s5.sh  s6.sh  s7.sh  s8.sh  s9.sh  v4.sh
chaitali@LAPTOP-UCEI0K1V:~$
```

4.ls -a
 ls -a intro
 Display, in succession, all of the available intro manual pages contained within the manual. It is
possible to quit between successive displays or skip any of them.



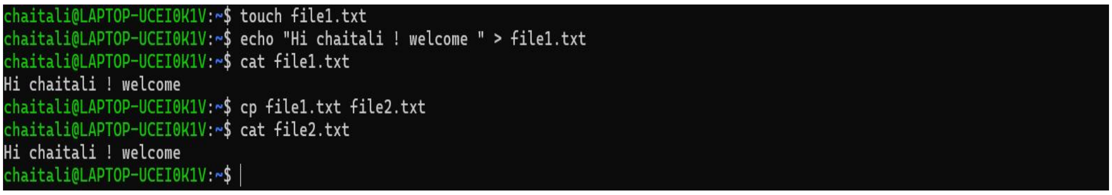
```
chaitali@LAPTOP-UCEI0K1V:~$ ls -a
.   .bash_history  .bashrc     .lesshst     .profile                   .viminfo         cdac      name.sh  s11.sh  s16.sh  s3.sh  s5.sh  s7.sh  s9.sh
..  .bash_logout   .landscape  .motd_shown  .sudo_as_admin_successful  LinuxAssignment  demo.txt  s10.sh   s12.sh  s2.txt  s4.sh  s6.sh  s8.sh  v4.sh
```

5.rm file.txt
- Delete the file.txt



```
LinuxAssignment  cdac  demo.txt  name.sh  s10.sh  s11.sh  s12.sh  s16.sh  s2.txt  s3.sh  s4.sh  s5.sh  s6.sh  s7.sh  s8.sh  s9.sh  v4.sh
chaitali@LAPTOP-UCEI0K1V:~$ rm s2.txt
chaitali@LAPTOP-UCEI0K1V:~$ ls
LinuxAssignment  cdac  demo.txt  name.sh  s10.sh  s11.sh  s12.sh  s16.sh  s3.sh  s4.sh  s5.sh  s6.sh  s7.sh  s8.sh  s9.sh  v4.sh
chaitali@LAPTOP-UCEI0K1V:~$ |
```

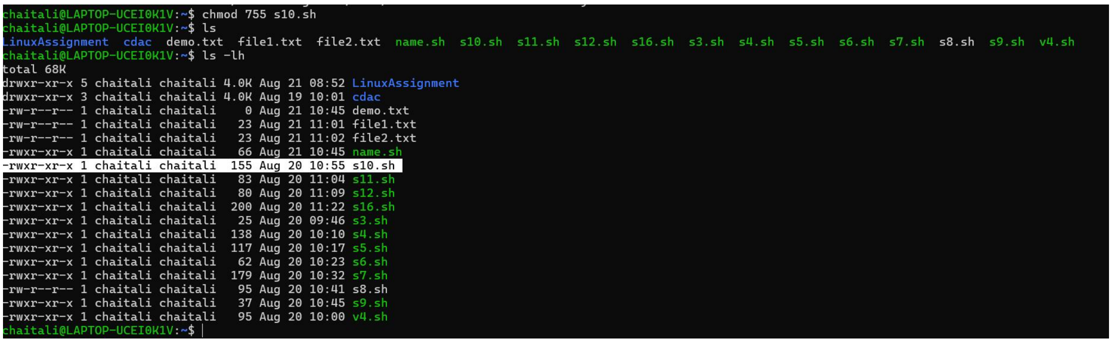
6.cp file1.txt file2.txt
-Copy the file1.txt to file2.txt



```
chaitali@LAPTOP-UCEI0K1V:~$ touch file1.txt
chaitali@LAPTOP-UCEI0K1V:~$ echo "Hi chaitali ! welcome " > file1.txt
chaitali@LAPTOP-UCEI0K1V:~$ cat file1.txt
Hi chaitali ! welcome
chaitali@LAPTOP-UCEI0K1V:~$ cp file1.txt file2.txt
chaitali@LAPTOP-UCEI0K1V:~$ cat file2.txt
Hi chaitali ! welcome
chaitali@LAPTOP-UCEI0K1V:~$ |
```

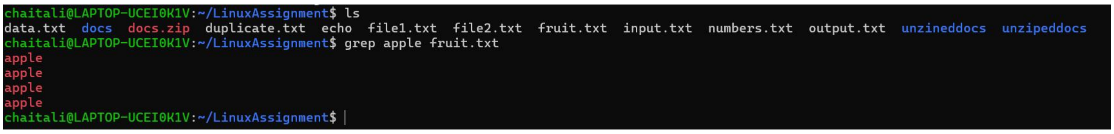
7.mv file.txt /path/to/directory/
-file.txt will be moved to the given path

8.chmod 755 script.sh
-Give permission read write execute.



```
chaitali@LAPTOP-UCEI0K1V:~$ chmod 755 s10.sh
chaitali@LAPTOP-UCEI0K1V:~$ ls
LinuxAssignment  cdac  demo.txt  file1.txt  file2.txt  name.sh  s10.sh  s11.sh  s12.sh  s16.sh  s3.sh  s4.sh  s5.sh  s6.sh  s7.sh  s8.sh  s9.sh  v4.sh
chaitali@LAPTOP-UCEI0K1V:~$ ls -lh
total 68K
drwxr-xr-x 5 chaitali chaitali 4.0K Aug 21 08:52 LinuxAssignment
drwxr-xr-x 3 chaitali chaitali 4.0K Aug 19 10:01 cdac
-rw-r--r-- 1 chaitali chaitali    0 Aug 21 10:45 demo.txt
-rw-r--r-- 1 chaitali chaitali   23 Aug 21 11:01 file1.txt
-rw-r--r-- 1 chaitali chaitali   23 Aug 21 11:02 file2.txt
-rwxr-xr-x 1 chaitali chaitali   66 Aug 21 10:45 name.sh
-rwxr-xr-x 1 chaitali chaitali  155 Aug 20 10:55 s10.sh
-rwxr-xr-x 1 chaitali chaitali   83 Aug 20 11:04 s11.sh
-rwxr-xr-x 1 chaitali chaitali   80 Aug 20 11:09 s12.sh
-rwxr-xr-x 1 chaitali chaitali  200 Aug 20 11:22 s16.sh
-rwxr-xr-x 1 chaitali chaitali   25 Aug 20 09:46 s3.sh
-rwxr-xr-x 1 chaitali chaitali  138 Aug 20 10:10 s4.sh
-rwxr-xr-x 1 chaitali chaitali  117 Aug 20 10:17 s5.sh
-rwxr-xr-x 1 chaitali chaitali   62 Aug 20 10:23 s6.sh
-rwxr-xr-x 1 chaitali chaitali  179 Aug 20 10:32 s7.sh
-rw-r--r-- 1 chaitali chaitali   95 Aug 20 10:41 s8.sh
-rwxr-xr-x 1 chaitali chaitali   37 Aug 20 10:45 s9.sh
-rwxr-xr-x 1 chaitali chaitali   95 Aug 20 10:00 v4.sh
chaitali@LAPTOP-UCEI0K1V:~$ |
```

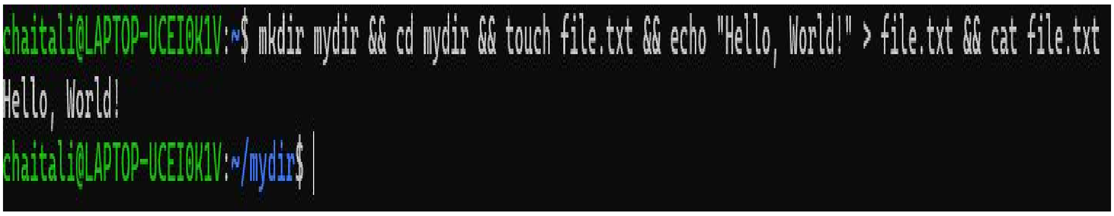
9.grep "pattern" file.txt
-searches word in the file and finds matching lines.



```
chaitali@LAPTOP-UCEI0K1V:~/LinuxAssignment$ ls
data.txt  docs  docs.zip  duplicate.txt  echo  file1.txt  file2.txt  fruit.txt  input.txt  numbers.txt  output.txt  unzineddocs  unzipeddocs
chaitali@LAPTOP-UCEI0K1V:~/LinuxAssignment$ grep apple fruit.txt
apple
apple
apple
apple
chaitali@LAPTOP-UCEI0K1V:~/LinuxAssignment$ |
```

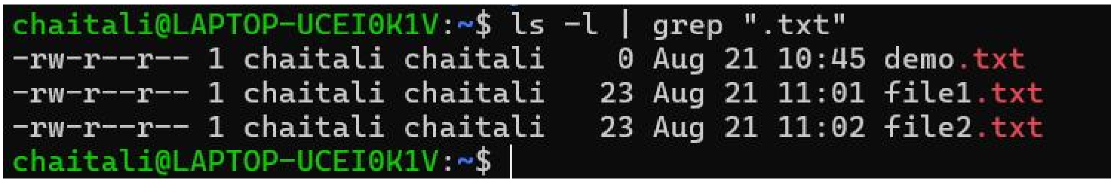
10.
kill PID
-kill the process with given process ID

11.mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
-Create a directory mydir
Change directory to mydir
Create an empty file file.txt
Write “hello world!” into file.txt
Display its contents



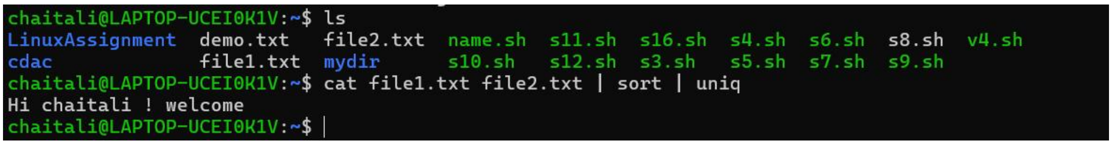
```
chaitali@LAPTOP-UCEI0K1V:~$ mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
Hello, World!
chaitali@LAPTOP-UCEI0K1V:~/mydir$
```

12.ls -l | grep ".txt"
- List files in long format and filters only .txt files



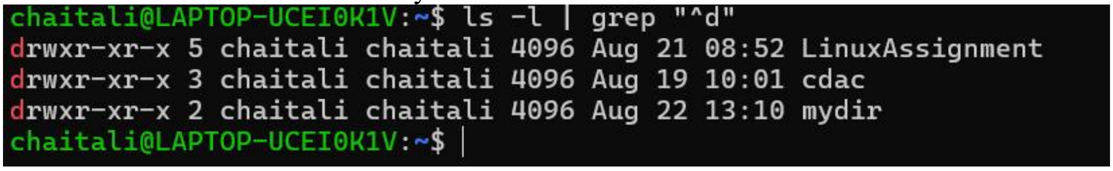
```
chaitali@LAPTOP-UCEI0K1V:~$ ls -l | grep ".txt"
-rw-r--r-- 1 chaitali chaitali    0 Aug 21 10:45 demo.txt
-rw-r--r-- 1 chaitali chaitali   23 Aug 21 11:01 file1.txt
-rw-r--r-- 1 chaitali chaitali   23 Aug 21 11:02 file2.txt
chaitali@LAPTOP-UCEI0K1V:~$
```

13.cat file1.txt file2.txt | sort | uniq
-concatinate content of file1.txt and file2.txt and sorts them by removing duplicate



```
chaitali@LAPTOP-UCEI0K1V:~$ ls
LinuxAssignment  demo.txt   file2.txt  name.sh  s11.sh  s16.sh  s4.sh  s6.sh  s8.sh  v4.sh
cdac             file1.txt  mydir      s10.sh   s12.sh  s3.sh   s5.sh  s7.sh  s9.sh
chaitali@LAPTOP-UCEI0K1V:~$ cat file1.txt file2.txt | sort | uniq
Hi chaitali ! welcome
chaitali@LAPTOP-UCEI0K1V:~$
```

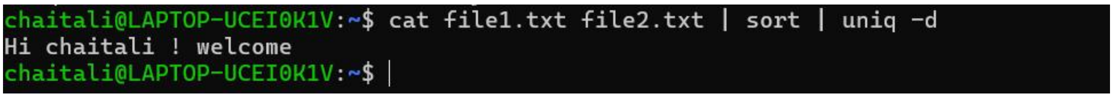
14.ls -l | grep "^d"
-Lists details of files and filters only directories.



```
chaitali@LAPTOP-UCEI0K1V:~$ ls -l | grep "^d"
drwxr-xr-x 5 chaitali chaitali 4096 Aug 21 08:52 LinuxAssignment
drwxr-xr-x 3 chaitali chaitali 4096 Aug 19 10:01 cdac
drwxr-xr-x 2 chaitali chaitali 4096 Aug 22 13:10 mydir
chaitali@LAPTOP-UCEI0K1V:~$
```

15.grep -r "pattern" /path/to/directory/
- searches “pattern” inside all files in the given directory

16.cat file1.txt file2.txt | sort | uniq –d
-prints only the duplicate lines common between two files



```
chaitali@LAPTOP-UCEI0K1V:~$ cat file1.txt file2.txt | sort | uniq -d
Hi chaitali ! welcome
chaitali@LAPTOP-UCEI0K1V:~$
```

17.chmod 644 file.txt
-sets permission

18cp -r source_directory destination_directory
-copie source directory and all its contents

19.find /path/to/search -name "*.txt"
-finds all .txt files under the path

20.chmod u+x file.txt
-gives the file’s user permission to execute

21.echo $PATH
-prints the system path env variable

PART -B

Identify True or False:

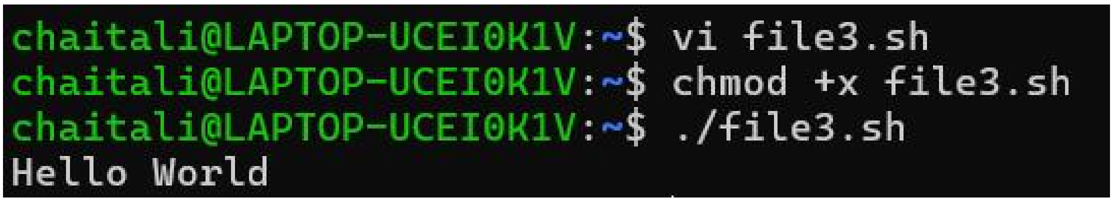
1.**ls** is used to list files and directories in a directory.- **True**
2.**mv** is used to move files and directories. **True**
3.cd is used to copy files and directories.**Flase**
4.pwd stands for "print working directory" and displays the current directory. **True**
5.grep is used to search for patterns 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.-**True**
8.rm -rf file.txt deletes a file forcefully without confirmation.**-False**

Identify the Incorrect Commands:

1.chmodx is used to change file permissions.-**chomd**
2.cpy is used to copy files and directories.-**cp**
3.mkfile is used to create a new file.**touch filename**
4.catx is used to concatenate files.-**cat**
5.rn is used to rename files.**-mv old newname**

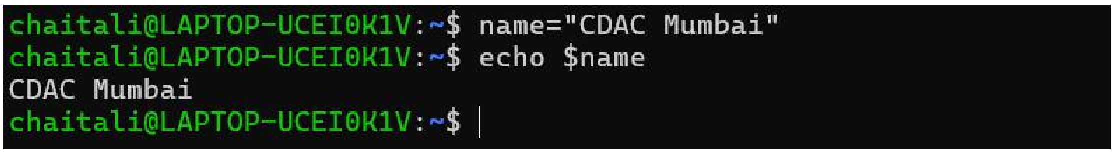
PART C

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



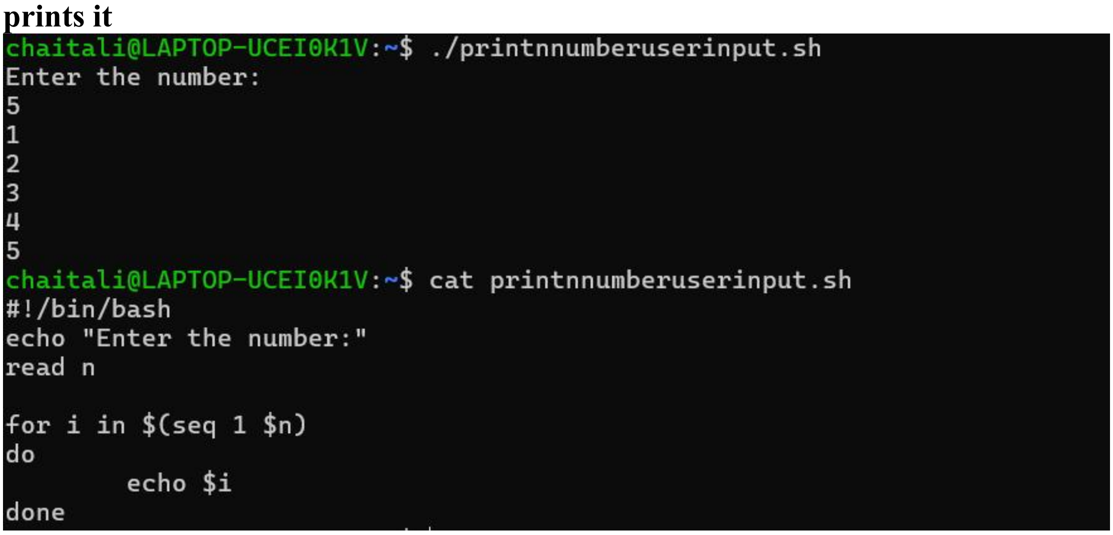
```
chaitali@LAPTOP-UCEI0K1V:~$ vi file3.sh
chaitali@LAPTOP-UCEI0K1V:~$ chmod +x file3.sh
chaitali@LAPTOP-UCEI0K1V:~$ ./file3.sh
Hello World
```

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



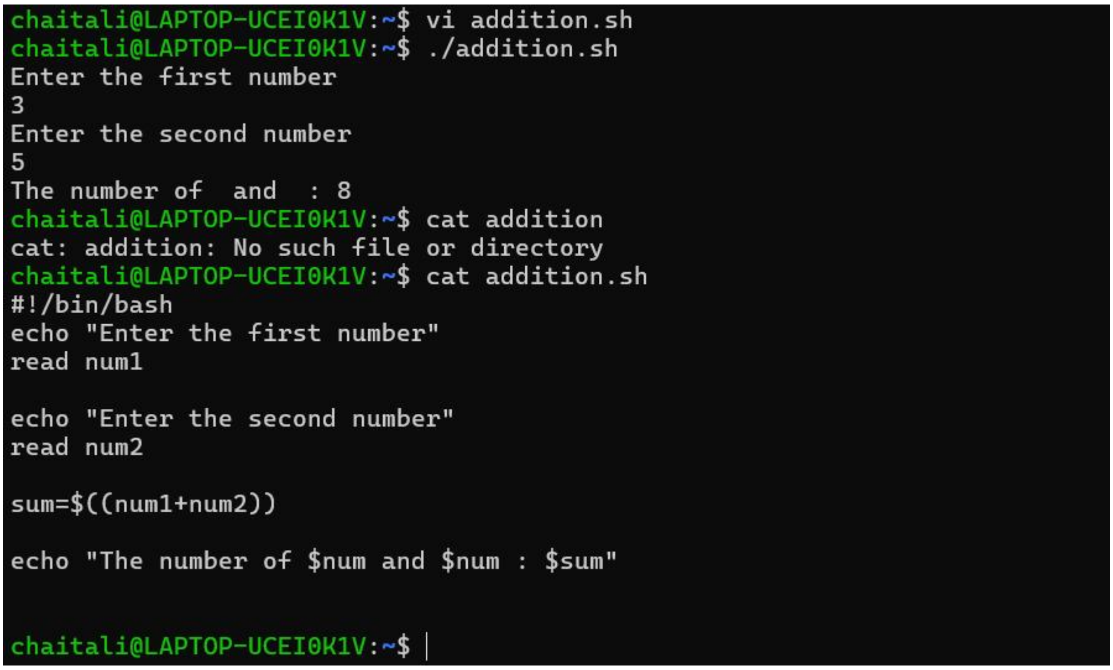
```
chaitali@LAPTOP-UCEI0K1V:~$ name="CDAC Mumbai"
chaitali@LAPTOP-UCEI0K1V:~$ echo $name
CDAC Mumbai
chaitali@LAPTOP-UCEI0K1V:~$ |
```

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



```
chaitali@LAPTOP-UCEI0K1V:~$ ./printnnumberuserinput.sh
Enter the number:
5
1
2
3
4
5
chaitali@LAPTOP-UCEI0K1V:~$ cat printnnumberuserinput.sh
#!/bin/bash
echo "Enter the number:"
read n

for i in $(seq 1 $n)
do
        echo $i
done
```

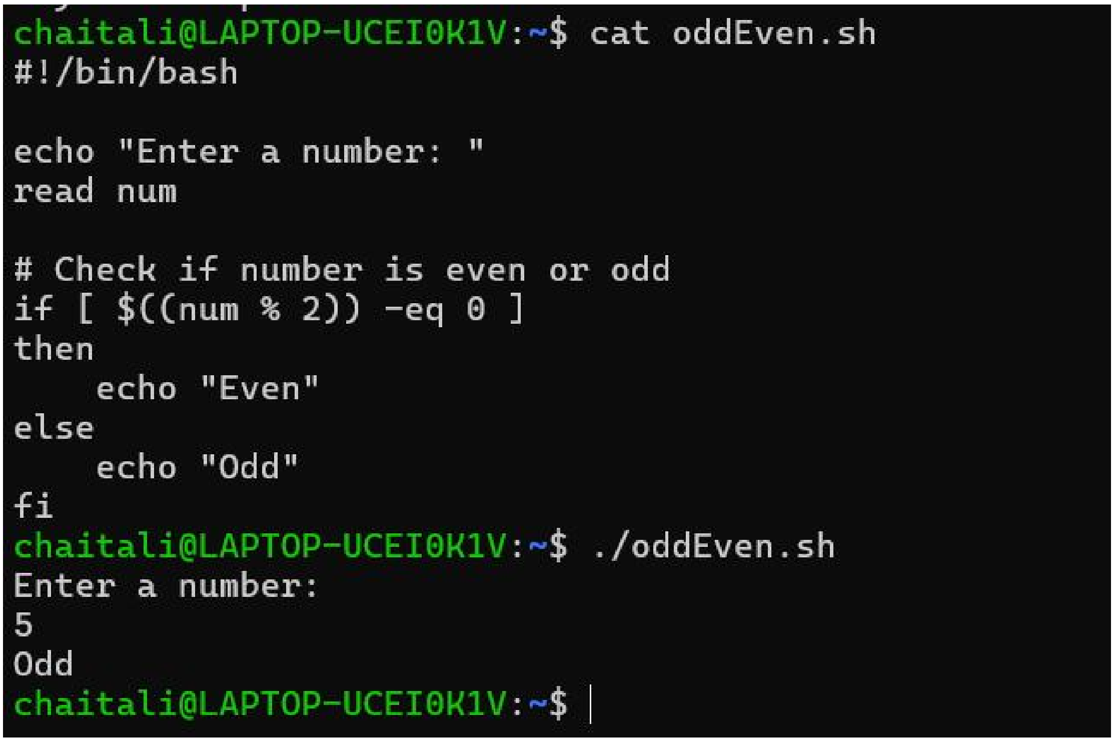
Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the
Result.



```
chaitali@LAPTOP-UCEI0K1V:~$ vi addition.sh
chaitali@LAPTOP-UCEI0K1V:~$ ./addition.sh
Enter the first number
3
Enter the second number
5
The number of  and  : 8
chaitali@LAPTOP-UCEI0K1V:~$ cat addition
cat: addition: No such file or directory
chaitali@LAPTOP-UCEI0K1V:~$ cat addition.sh
#!/bin/bash
echo "Enter the first number"
read num1

echo "Enter the second number"
read num2

sum=$((num1+num2))

echo "The number of $num and $num : $sum"


chaitali@LAPTOP-UCEI0K1V:~$ |
```

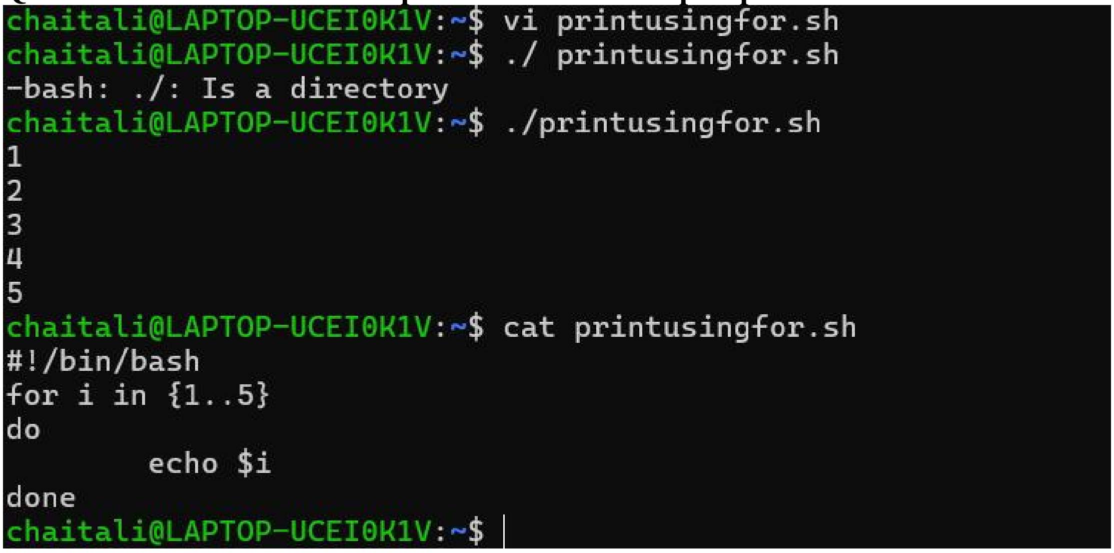
Question 5: Write a shell script that takes a number as input and prints "Even" if it is even,

otherwise prints "Odd".



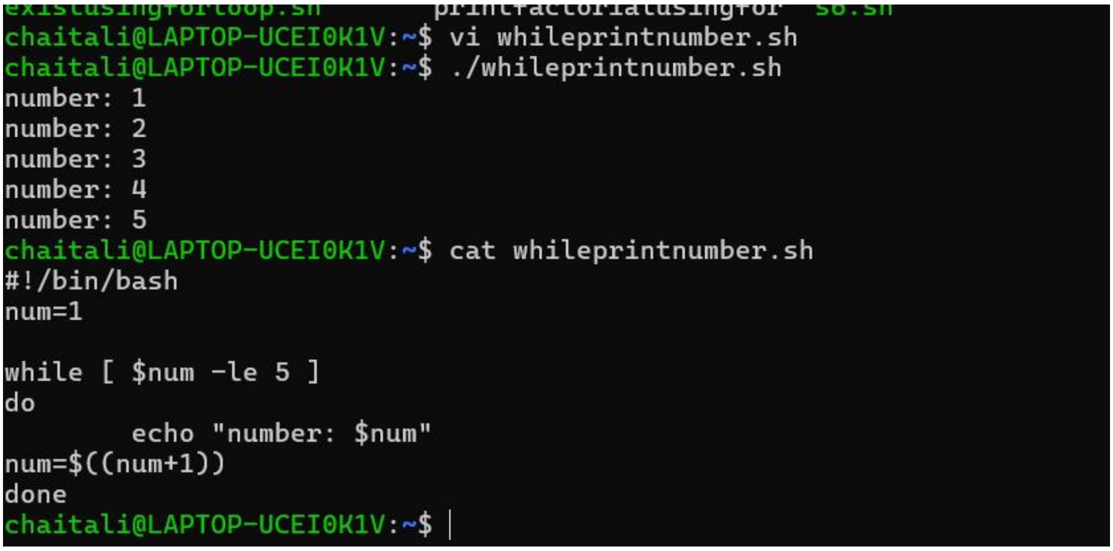
```
chaitali@LAPTOP-UCEI0K1V:~$ cat oddEven.sh
#!/bin/bash

echo "Enter a number: "
read num

# Check if number is even or odd
if [ $((num % 2)) -eq 0 ]
then
    echo "Even"
else
    echo "Odd"
fi
chaitali@LAPTOP-UCEI0K1V:~$ ./oddEven.sh
Enter a number:
5
Odd
chaitali@LAPTOP-UCEI0K1V:~$
```

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



```
chaitali@LAPTOP-UCEI0K1V:~$ vi printusingfor.sh
chaitali@LAPTOP-UCEI0K1V:~$ ./ printusingfor.sh
-bash: ./: Is a directory
chaitali@LAPTOP-UCEI0K1V:~$ ./printusingfor.sh
1
2
3
4
5
chaitali@LAPTOP-UCEI0K1V:~$ cat printusingfor.sh
#!/bin/bash
for i in {1..5}
do
        echo $i
done
chaitali@LAPTOP-UCEI0K1V:~$
```

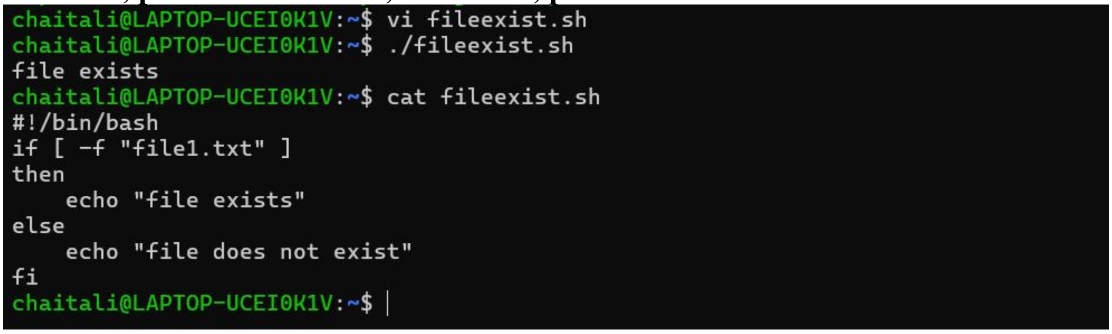
Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.



```
existusingforloop.sh        printfactorialusingfor  s8.sh
chaitali@LAPTOP-UCEI0K1V:~$ vi whileprintnumber.sh
chaitali@LAPTOP-UCEI0K1V:~$ ./whileprintnumber.sh
number: 1
number: 2
number: 3
number: 4
number: 5
chaitali@LAPTOP-UCEI0K1V:~$ cat whileprintnumber.sh
#!/bin/bash
num=1

while [ $num -le 5 ]
do
        echo "number: $num"
num=$((num+1))
done
chaitali@LAPTOP-UCEI0K1V:~$ |
```

Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory.

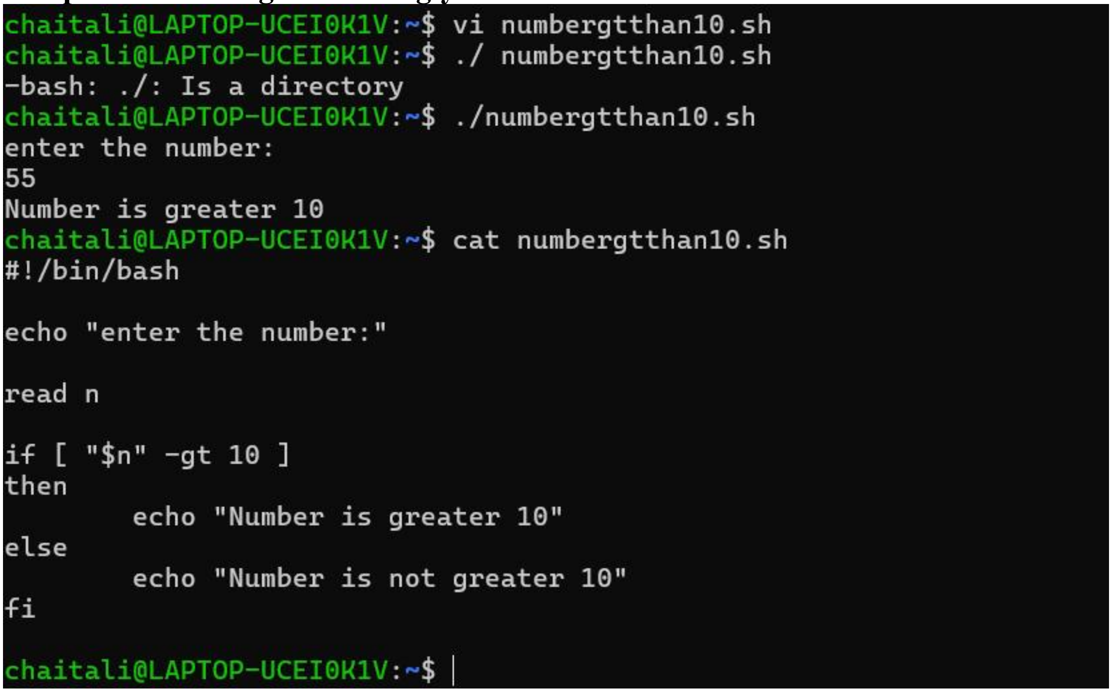
If it does, print "File exists", otherwise, print "File does not exist".



```
chaitali@LAPTOP-UCEI0K1V:~$ vi fileexist.sh
chaitali@LAPTOP-UCEI0K1V:~$ ./fileexist.sh
file exists
chaitali@LAPTOP-UCEI0K1V:~$ cat fileexist.sh
#!/bin/bash
if [ -f "file1.txt" ]
then
    echo "file exists"
else
    echo "file does not exist"
fi
chaitali@LAPTOP-UCEI0K1V:~$ |
```

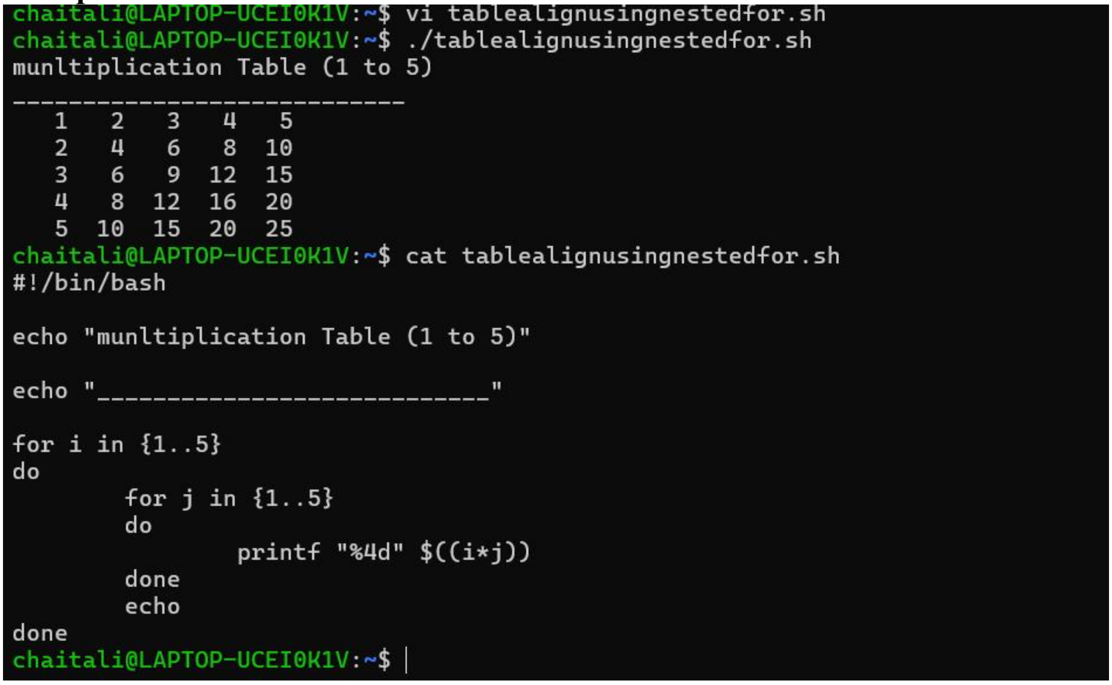
Question 9: Write a shell script that uses the if statement to check if a number is greater than 10

and prints a message accordingly.



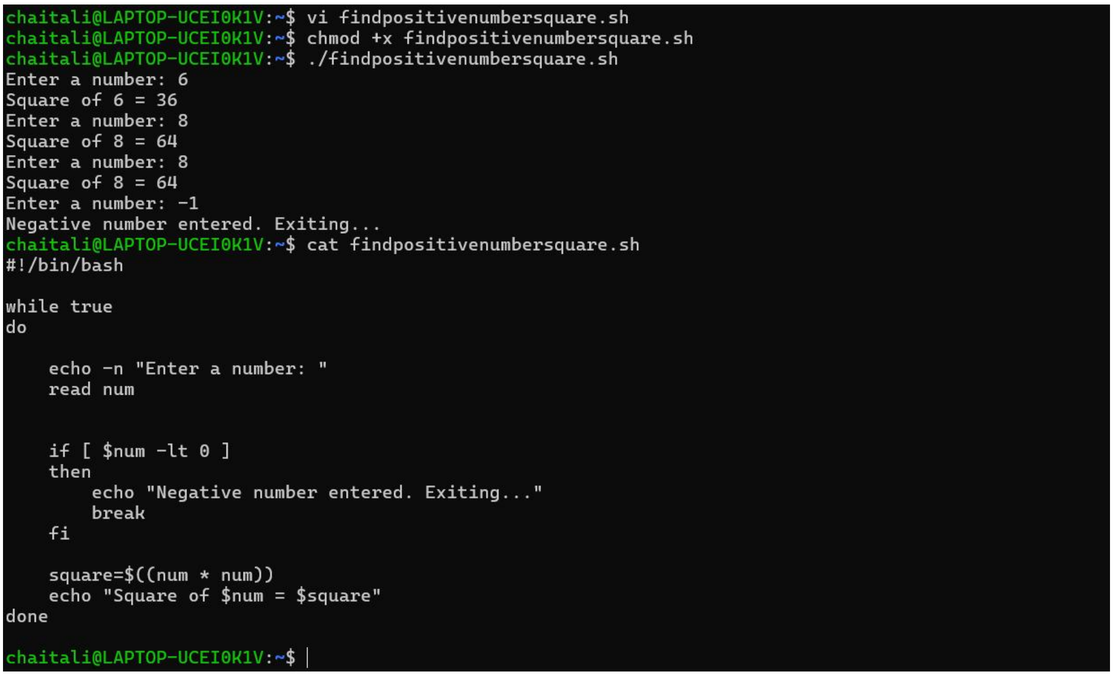
```
chaitali@LAPTOP-UCEI0K1V:~$ vi numbergtthan10.sh
chaitali@LAPTOP-UCEI0K1V:~$ ./ numbergtthan10.sh
-bash: ./: Is a directory
chaitali@LAPTOP-UCEI0K1V:~$ ./numbergtthan10.sh
enter the number:
55
Number is greater 10
chaitali@LAPTOP-UCEI0K1V:~$ cat numbergtthan10.sh
#!/bin/bash

echo "enter the number:"

read n

if [ "$n" -gt 10 ]
then
        echo "Number is greater 10"
else
        echo "Number is not greater 10"
fi

chaitali@LAPTOP-UCEI0K1V:~$ |
```

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.



```
chaitali@LAPTOP-UCEI0K1V:~$ vi tablealignusingnestedfor.sh
chaitali@LAPTOP-UCEI0K1V:~$ ./tablealignusingnestedfor.sh
munltiplication Table (1 to 5)

----------------------------
   1   2   3   4   5
   2   4   6   8  10
   3   6   9  12  15
   4   8  12  16  20
   5  10  15  20  25
chaitali@LAPTOP-UCEI0K1V:~$ cat tablealignusingnestedfor.sh
#!/bin/bash

echo "munltiplication Table (1 to 5)"

echo "____________________________"

for i in {1..5}
do
        for j in {1..5}
        do
                printf "%4d" $((i*j))
        done
        echo
done
chaitali@LAPTOP-UCEI0K1V:~$ |
```

Question 11: 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.



```
chaitali@LAPTOP-UCEI0K1V:~$ vi findpositivenumbersquare.sh
chaitali@LAPTOP-UCEI0K1V:~$ chmod +x findpositivenumbersquare.sh
chaitali@LAPTOP-UCEI0K1V:~$ ./findpositivenumbersquare.sh
Enter a number: 6
Square of 6 = 36
Enter a number: 8
Square of 8 = 64
Enter a number: 8
Square of 8 = 64
Enter a number: -1
Negative number entered. Exiting...
chaitali@LAPTOP-UCEI0K1V:~$ cat findpositivenumbersquare.sh
#!/bin/bash

while true
do

    echo -n "Enter a number: "
    read num


    if [ $num -lt 0 ]
    then
        echo "Negative number entered. Exiting..."
        break
    fi

    square=$((num * num))
    echo "Square of $num = $square"
done

chaitali@LAPTOP-UCEI0K1V:~$
```