Assignment 2
Section A
What will the following commands do?

- **echo "Hello, World!"**
- **name="Productive"**
- **touch file.txt**
- **ls -a**
- **rm file.txt**
- **cp file1.txt file2.txt**
- **mv file.txt /path/to/directory/**
- **chmod 755 script.sh**
- **grep "pattern" file.txt**
- **kill PID**
- **mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt**
- **ls -l | grep ".txt"**
- **cat file1.txt file2.txt | sort | uniq**
- **ls -l | grep "^d"**
- **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**

 echo $PATH

Part B

Identify True or False:

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

2. mv is used to move files and directories.

3. cd is used to copy files and directories.

4. pwd stands for "print working directory" and displays the current directory.

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

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

permissions to group and others.

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.

Identify the Incorrect Commands:

1. chmodx is used to change file permissions.

2. cpy is used to copy files and directories.

3. mkfile is used to create a new file.

4. catx is used to concatenate files.

5. rn is used to rename files.

Part E

1. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	6

Calculate the average waiting time using First-Come, First-Served (FCFS) scheduling.

2. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	3
P2	1	5
P3	2	1
P4	3	4

Calculate the average turnaround time using Shortest Job First (SJF) scheduling.

3. Consider the following processes with arrival times, burst times, and priorities (lower number

indicates higher priority):

Process	Arrival Time	Burst Time	Priority
P1	0	6	3
P2	1	4	1
P3	2	7	4

P4	**3**	**2**	**2**

Calculate the average waiting time using Priority Scheduling.

4. Consider the following processes with arrival times and burst times, and the time quantum for

Round Robin scheduling is 2 units:

Process	**Arrival Time**	**Burst Time**
P1	**0**	**4**
P2	**1**	**5**
P3	**2**	**2**
P4	**3**	**3**

Calculate the average turnaround time using Round Robin scheduling.

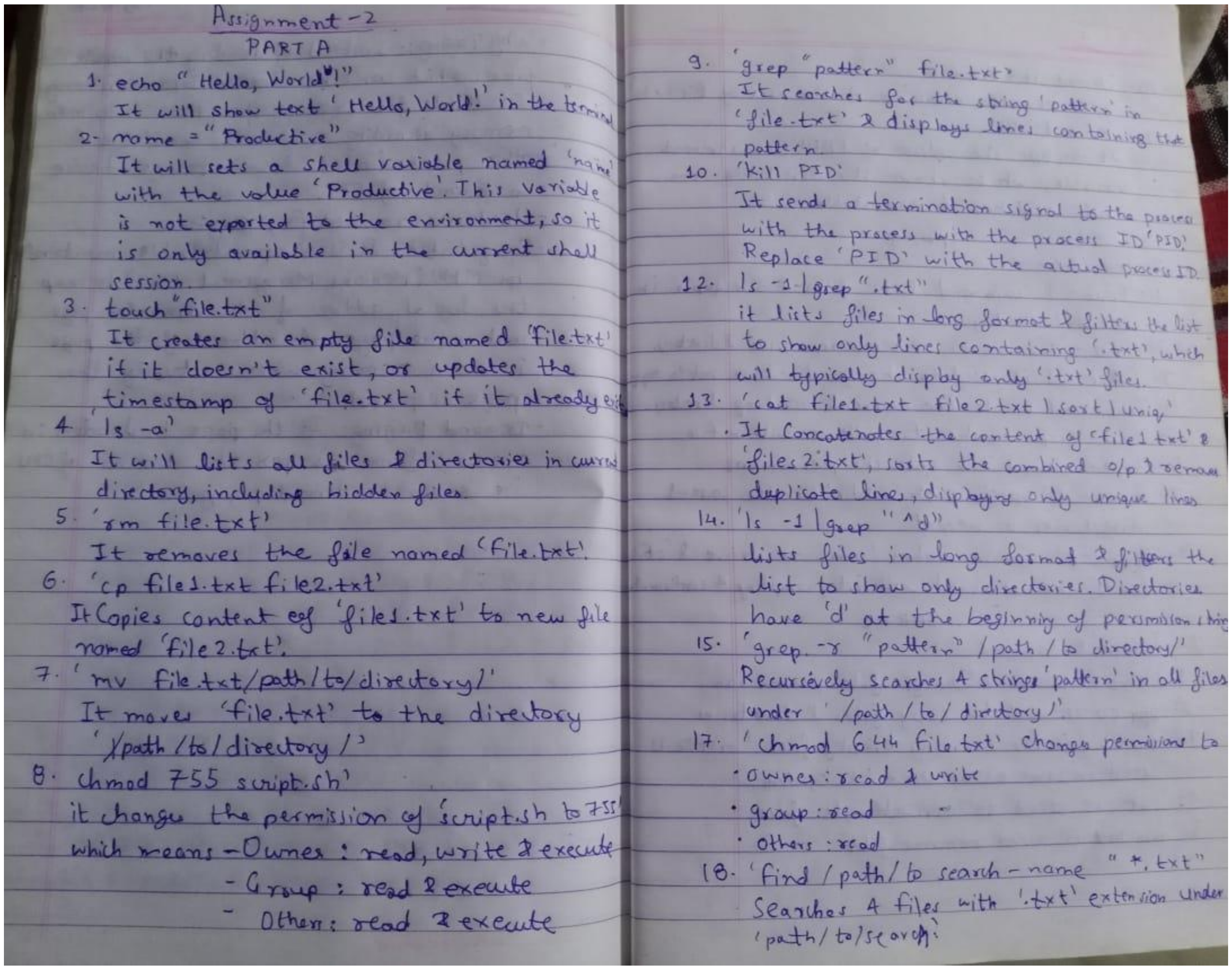
5. Consider a program that uses the fork() system call to create a child process. Initially, the parent

process has a variable x with a value of 5. After forking, both the parent and child processes

increment the value of x by 1.

What will be the final values of x in the parent and child processes after the fork() call?

Picture of Solution of Part A, B and E

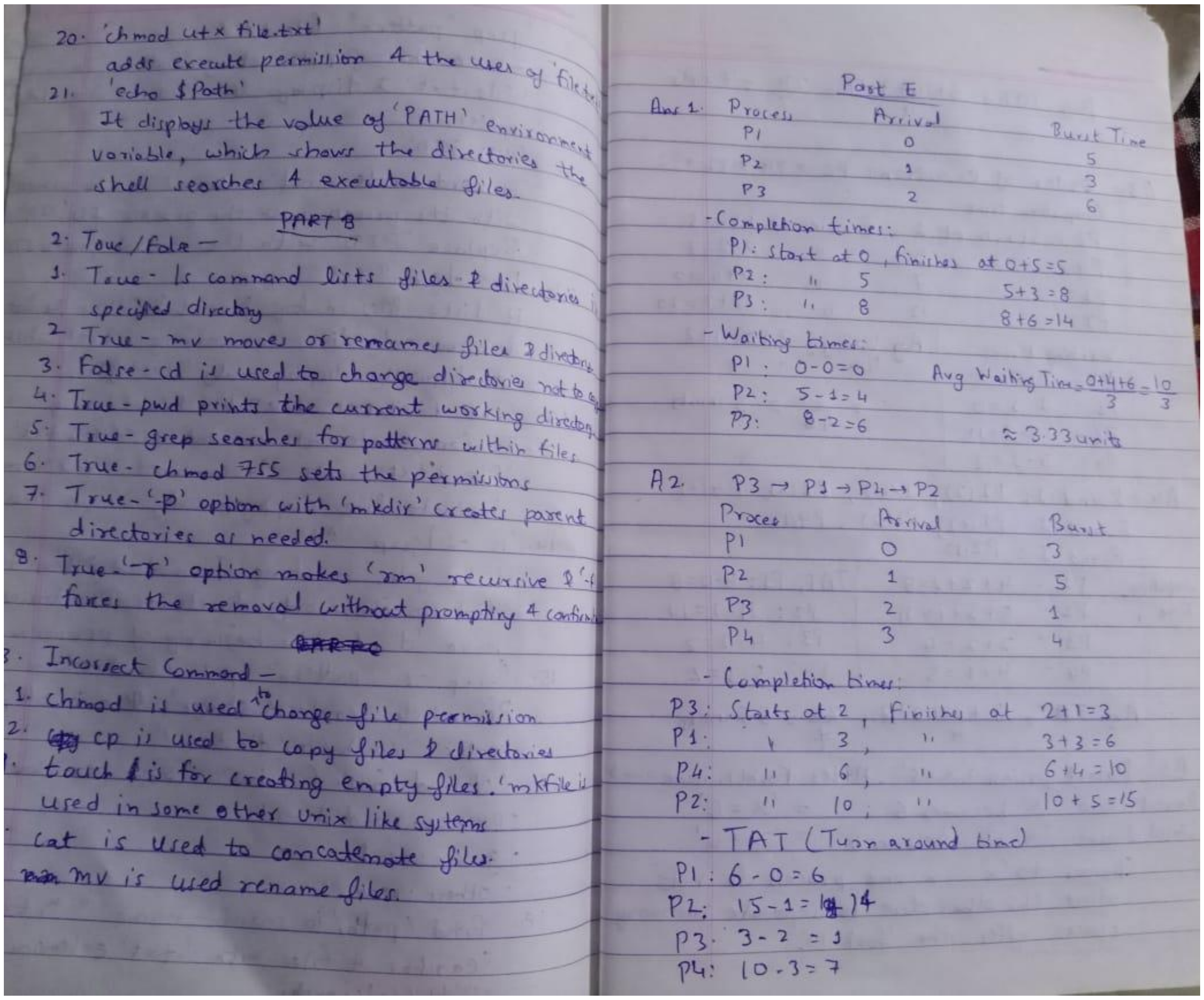


Assignment -2

PART A

1. echo " Hello, World!"
   It will show text ' Hello, World!' in the terminal
2. name = " Productive"
   It will sets a shell variable named 'name' with the value 'Productive'. This variable is not exported to the environment, so it is only available in the current shell session.
3. touch "file.txt"
   It creates an empty file named 'file.txt' if it doesn't exist, or updates the timestamp of 'file.txt' if it already exists
4. 'ls -a'
   It will lists all files & directories in current directory, including hidden files.
5. 'rm file.txt'
   It removes the file named 'file.txt'.
6. 'cp file1.txt file2.txt'
   It Copies content of 'file1.txt' to new file named 'file2.txt'.
7. ' mv file.txt/path/to/directory/'
   It moves 'file.txt' to the directory '/path/to/directory/'
8. chmod 755 script.sh'
   it changes the permission of 'script.sh to 755' which means - Owner : read, write & execute
   - Group : read & execute
   - Others: read & execute

9. 'grep "pattern" file.txt'
   It searches for the string 'pattern' in 'file.txt' & displays lines containing that pattern.
10. 'kill PID'
    It sends a termination signal to the process with the process with the process ID 'PID'. Replace 'PID' with the actual process ID.
12. ls -1 | grep ".txt"
    it lists files in long format & filters the list to show only lines containing '.txt', which will typically display only '.txt' files.
13. 'cat file1.txt file2.txt | sort | uniq'
    . It Concatenates the content of 'file1.txt' & 'files2.txt', sorts the combined o/p & removes duplicate lines, displaying only unique lines
14. 'ls -1 | grep "^d"
    lists files in long format & filters the list to show only directories. Directories have 'd' at the beginning of permission string
15. 'grep -r "pattern" /path/to/directory/'
    Recursively searches 4 strings 'pattern' in all files under '/path/to/directory/'.
17. 'chmod 644 file.txt' changes permissions to
    - owner : read & write
    - group : read
    - others : read
18. 'find /path/to search -name "*.txt"
    Searches 4 files with '.txt' extension under 'path/to/search'.



20. 'chmod u+x file.txt'
adds execute permission 4 the user of 'file.t

21. 'echo $Path'
It displays the value of 'PATH' environment variable, which shows the directories the shell searches 4 executable files.

PART B

2. True / False —

1. True - ls command lists files & directories i specified directory
2. True - mv moves or renames files & directories
3. False - cd is used to change directories not to c
4. True - pwd prints the current working directory
5. True - grep searches for patterns within files
6. True - chmod 755 sets the permissions
7. True - '-p' option with 'mkdir' creates parent directories as needed.
8. True - '-r' option makes 'rm' recursive & '-f forces the removal without prompting 4 confirm

~~[illegible]~~

3. Incorrect Command —

1. chmod is used to change file permission
2. ~~[illegible]~~ cp is used to copy files & directories
1. touch ~~[illegible]~~ is for creating empty files. 'mkfile is used in some other unix like systems
. cat is used to concatenate files.
~~[illegible]~~ mv is used rename files.

Part E

Ans 1.

Process	Arrival	Burst Time
P1	0	5
P2	1	3
P3	2	6

- Completion times:
P1: start at 0, finishes at 0+5=5
P2: " 5 5+3=8
P3: " 8 8+6=14

- Waiting times:
P1: 0-0=0
P2: 5-1=4
P3: 8-2=6

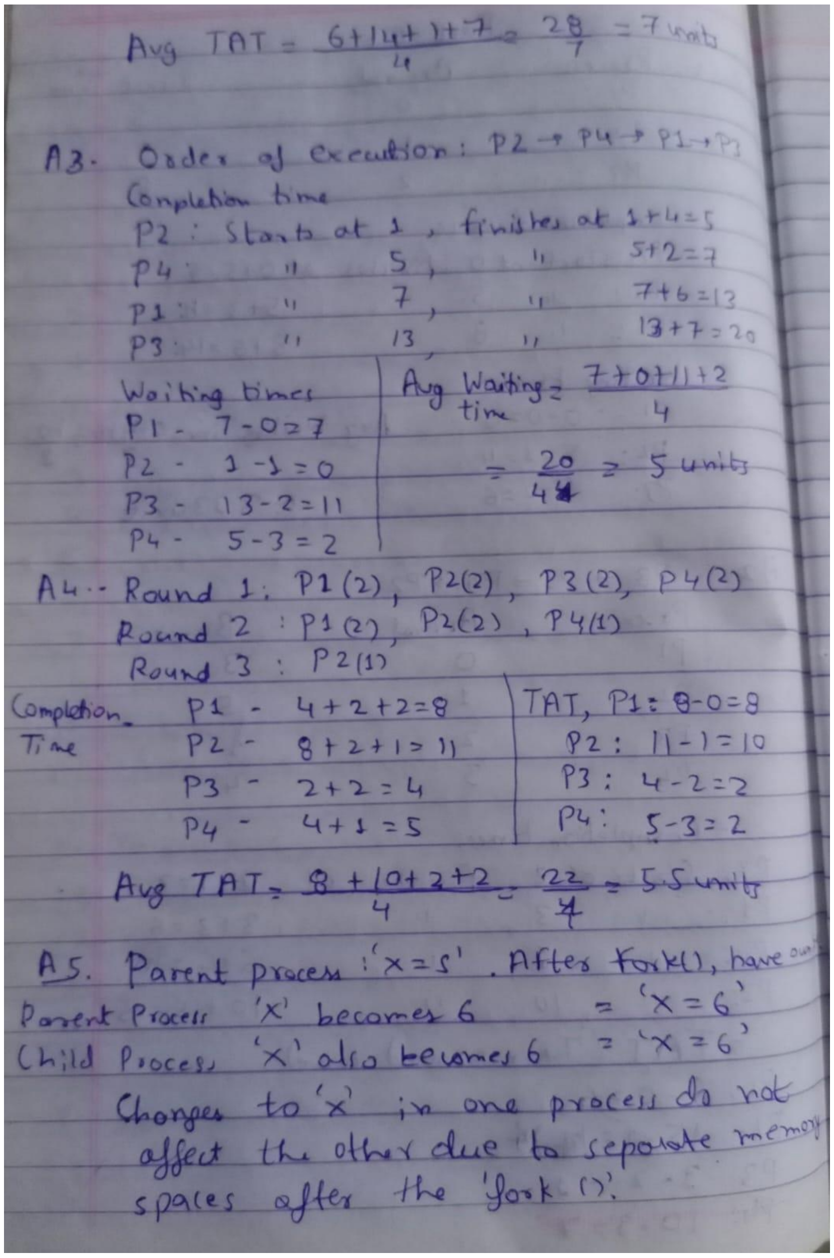
Avg Waiting Time $= \frac{0+4+6}{3} = \frac{10}{3}$
≈ 3.33 units

A2. $P3 \rightarrow P1 \rightarrow P4 \rightarrow P2$

Process	Arrival	Burst
P1	0	3
P2	1	5
P3	2	1
P4	3	4

- Completion times:
P3: Starts at 2, finishes at 2+1=3
P1: " 3, " 3+3=6
P4: " 6, " 6+4=10
P2: " 10, " 10+5=15

- TAT (Turn around time)
P1: 6-0=6
P2: 15-1=~~[illegible]~~ 14
P3: 3-2 = 1
P4: 10-3=7



Avg TAT = $\frac{6+14+1+7}{4} = \frac{28}{7}$ = 7 units

A3. Order of execution: P2 → P4 → P1 → P3

Completion time

P2 :	Starts at	1,	finishes at	1+4=5
P4 :	"	5,	"	5+2=7
P1 :	"	7,	"	7+6=13
P3 :	"	13,	"	13+7=20

Waiting times

P1 - 7-0=7

P2 - 1-1=0

P3 - 13-2=11

P4 - 5-3=2

Avg Waiting time = $\frac{7+0+11+2}{4}$

= $\frac{20}{4}$ = 5 units

A4. Round 1: P1 (2), P2(2), P3(2), P4(2)

Round 2 : P1 (2), P2(2), P4(1)

Round 3 : P2 (1)

Completion Time	P1 - 4+2+2=8	TAT,	P1: 8-0=8
	P2 - 8+2+1=11		P2 : 11-1=10
	P3 - 2+2=4		P3 : 4-2=2
	P4 - 4+1=5		P4 : 5-3=2

Avg TAT = $\frac{8+10+2+2}{4} = \frac{22}{4}$ = 5.5 units

A5. Parent process : 'x=5'. After fork(), have own

Parent Process 'x' becomes 6 = 'x=6'

Child Process 'x' also becomes 6 = 'x=6'

Changes to 'x' in one process do not affect the other due to separate memory spaces after the 'fork ()'.

Part C

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

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

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

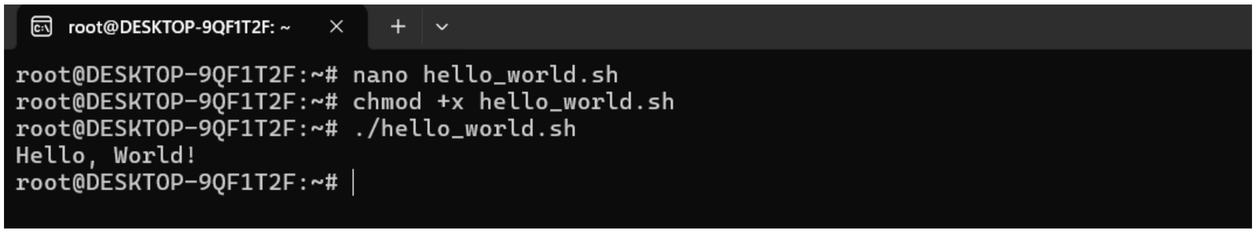
Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the

result.

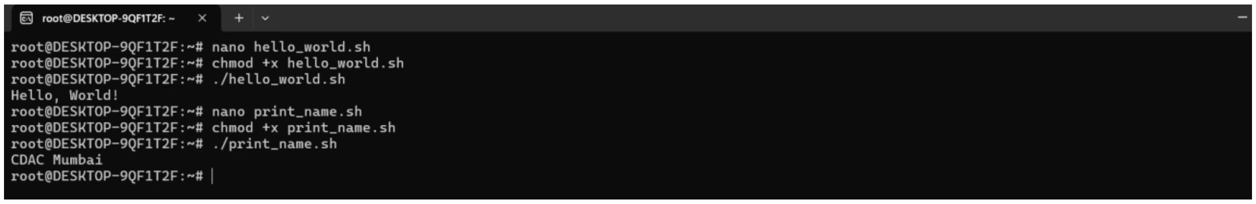
Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

Screenshots of the commands

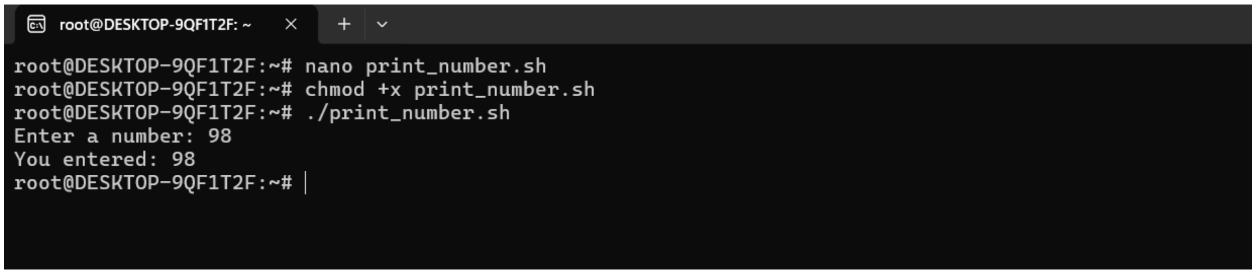
C1, C2, C3, C4, C5



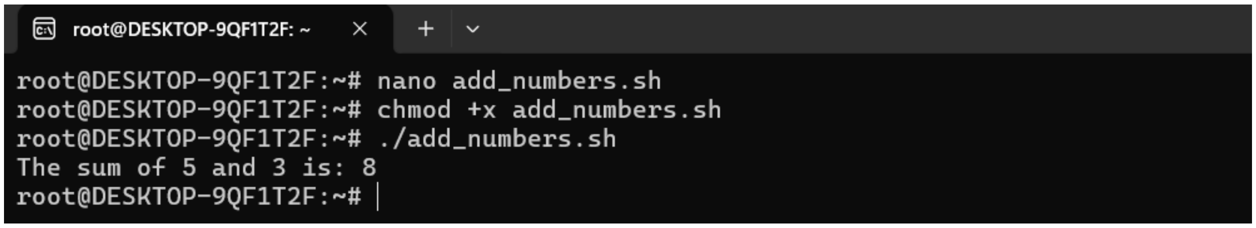
```
root@DESKTOP-9QF1T2F:~# nano hello_world.sh
root@DESKTOP-9QF1T2F:~# chmod +x hello_world.sh
root@DESKTOP-9QF1T2F:~# ./hello_world.sh
Hello, World!
root@DESKTOP-9QF1T2F:~#
```



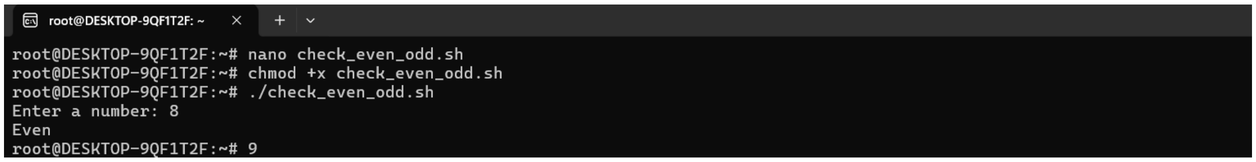
```
root@DESKTOP-9QF1T2F:~# nano hello_world.sh
root@DESKTOP-9QF1T2F:~# chmod +x hello_world.sh
root@DESKTOP-9QF1T2F:~# ./hello_world.sh
Hello, World!
root@DESKTOP-9QF1T2F:~# nano print_name.sh
root@DESKTOP-9QF1T2F:~# chmod +x print_name.sh
root@DESKTOP-9QF1T2F:~# ./print_name.sh
CDAC Mumbai
root@DESKTOP-9QF1T2F:~#
```



```
root@DESKTOP-9QF1T2F:~# nano print_number.sh
root@DESKTOP-9QF1T2F:~# chmod +x print_number.sh
root@DESKTOP-9QF1T2F:~# ./print_number.sh
Enter a number: 98
You entered: 98
root@DESKTOP-9QF1T2F:~#
```

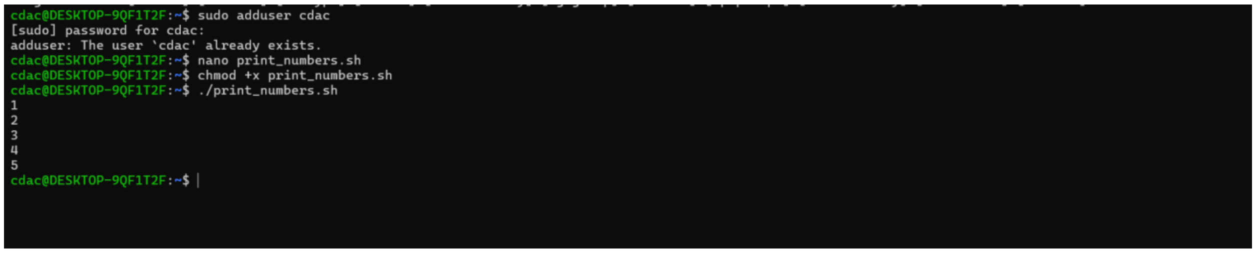


```
root@DESKTOP-9QF1T2F:~# nano add_numbers.sh
root@DESKTOP-9QF1T2F:~# chmod +x add_numbers.sh
root@DESKTOP-9QF1T2F:~# ./add_numbers.sh
The sum of 5 and 3 is: 8
root@DESKTOP-9QF1T2F:~#
```



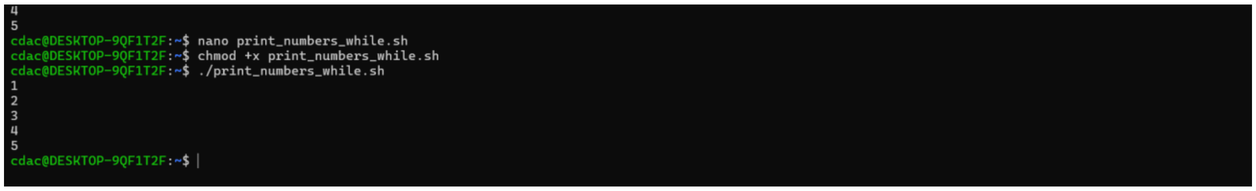
```
root@DESKTOP-9QF1T2F:~# nano check_even_odd.sh
root@DESKTOP-9QF1T2F:~# chmod +x check_even_odd.sh
root@DESKTOP-9QF1T2F:~# ./check_even_odd.sh
Enter a number: 8
Even
root@DESKTOP-9QF1T2F:~# 9
```

Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.
C6



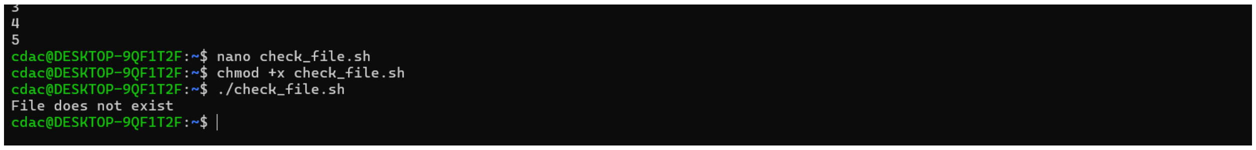
```
cdac@DESKTOP-9QF1T2F:~$ sudo adduser cdac
[sudo] password for cdac:
adduser: The user `cdac' already exists.
cdac@DESKTOP-9QF1T2F:~$ nano print_numbers.sh
cdac@DESKTOP-9QF1T2F:~$ chmod +x print_numbers.sh
cdac@DESKTOP-9QF1T2F:~$ ./print_numbers.sh
1
2
3
4
5
cdac@DESKTOP-9QF1T2F:~$ |
```

Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.
C7



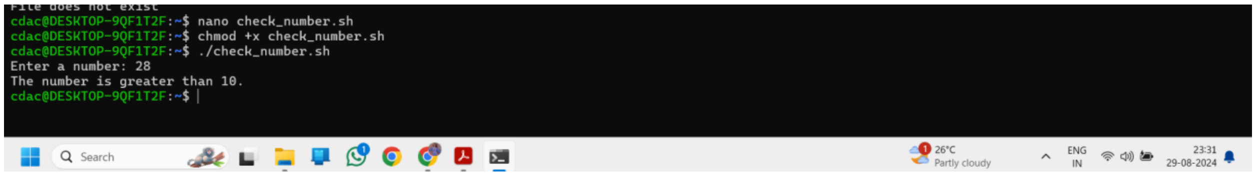
```
4
5
cdac@DESKTOP-9QF1T2F:~$ nano print_numbers_while.sh
cdac@DESKTOP-9QF1T2F:~$ chmod +x print_numbers_while.sh
cdac@DESKTOP-9QF1T2F:~$ ./print_numbers_while.sh
1
2
3
4
5
cdac@DESKTOP-9QF1T2F:~$ |
```

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".
C8



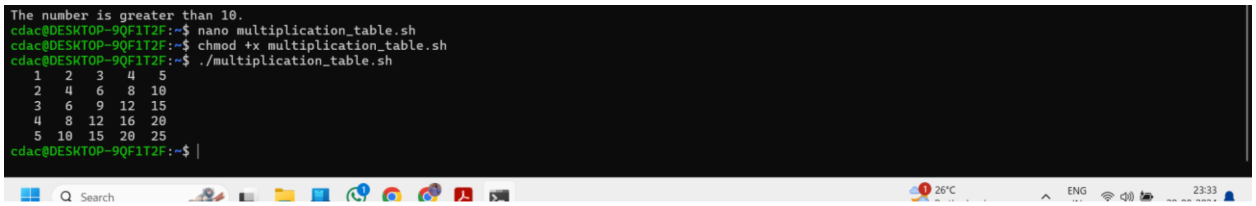
```
3
4
5
cdac@DESKTOP-9QF1T2F:~$ nano check_file.sh
cdac@DESKTOP-9QF1T2F:~$ chmod +x check_file.sh
cdac@DESKTOP-9QF1T2F:~$ ./check_file.sh
File does not exist
cdac@DESKTOP-9QF1T2F:~$ |
```

Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and a message accordingly
C9



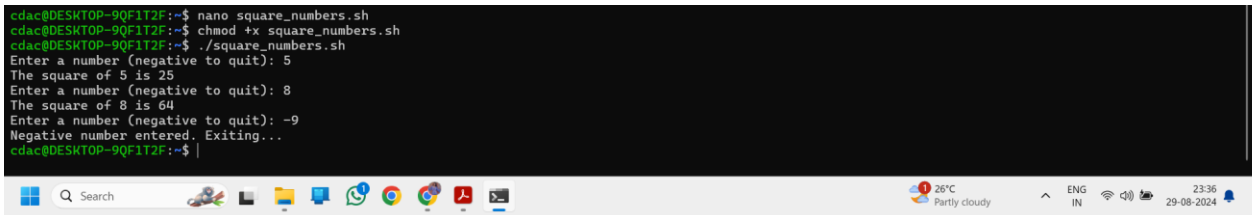
```
File does not exist
cdac@DESKTOP-9QF1T2F:~$ nano check_number.sh
cdac@DESKTOP-9QF1T2F:~$ chmod +x check_number.sh
cdac@DESKTOP-9QF1T2F:~$ ./check_number.sh
Enter a number: 28
The number is greater than 10.
cdac@DESKTOP-9QF1T2F:~$ |
```

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.
C10



```
The number is greater than 10.
cdac@DESKTOP-9QF1T2F:~$ nano multiplication_table.sh
cdac@DESKTOP-9QF1T2F:~$ chmod +x multiplication_table.sh
cdac@DESKTOP-9QF1T2F:~$ ./multiplication_table.sh
   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
cdac@DESKTOP-9QF1T2F:~$ |
```

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.
C11



```
cdac@DESKTOP-9QF1T2F:~$ nano square_numbers.sh
cdac@DESKTOP-9QF1T2F:~$ chmod +x square_numbers.sh
cdac@DESKTOP-9QF1T2F:~$ ./square_numbers.sh
Enter a number (negative to quit): 5
The square of 5 is 25
Enter a number (negative to quit): 8
The square of 8 is 64
Enter a number (negative to quit): -9
Negative number entered. Exiting...
cdac@DESKTOP-9QF1T2F:~$ |
```