ARPIT SHARMA

COS MODULE

PART-A

1. **echo "Hello, World!"**
    - Prints "Hello, World!" to the terminal.
2. **name="Productive"**
    - Assigns the value "Productive" to the variable name (only available in the current shell session).
3. **touch file.txt**
    - Creates an empty file named file.txt (or updates its timestamp if it already exists).
4. **ls -a**
    - Lists all files and directories, **including hidden files** (which start with .).
5. **rm file.txt**
    - Deletes file.txt permanently.
6. **cp file1.txt file2.txt**
    - Copies file1.txt to file2.txt (overwriting if file2.txt exists).
7. **mv file.txt /path/to/directory/**
    - Moves file.txt to the specified directory.
8. **chmod 755 script.sh**
    - Changes permissions of script.sh to:
        - **Owner:** Read, write, execute (7)
        - **Group & Others:** Read and execute (5)
9. **grep "pattern" file.txt**
    - Searches for "pattern" inside file.txt and prints matching lines.
10. **kill PID**
    - Terminates a process with the given PID (Process ID).
11. **mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt**
    - Creates a directory mydir, moves into it, creates file.txt, writes "Hello, World!" into it, and then displays its content.
12. **ls -l | grep ".txt"**
    - Lists files in long format (ls -l) and filters only .txt files.
13. **cat file1.txt file2.txt | sort | uniq**
    - Combines file1.txt and file2.txt, sorts them, and removes duplicate lines.

14. **ls -l | grep "^d"**
    - Lists only directories (lines starting with d in ls -l output).
15. **grep -r "pattern" /path/to/directory/**
    - Recursively searches for "pattern" in all files under /path/to/directory/.
16. **cat file1.txt file2.txt | sort | uniq –d**
    - Finds and prints only the **duplicate** lines from both files.
17. **chmod 644 file.txt**
    - Sets permissions:
        - **Owner:** Read and write (6)
        - **Group & Others:** Read-only (4)
18. **cp -r source_directory destination_directory**
    - Copies a directory (source_directory) and its contents **recursively**.
19. **find /path/to/search -name "*.txt"**
    - Searches for all .txt files in /path/to/search.
20. **chmod u+x file.txt**
    - Gives the **user (owner)** execute permission for file.txt.
21. **echo $PATH**
    - Displays the directories where the system looks for executable programs.

PART-2

• ls is used to list files and directories in a directory. – True

• mv is used to move files and directories. – True

• cd is used to copy files and directories. – False

• pwd stands for "print working directory" and displays the current directory. – True

• grep is used to search for patterns in files. – True

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

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

- rm -rf file.txt deletes a file forcefully without confirmation. – False

PART -3

1.chmod command is used to change file permissions.

2. cp command is used to copy files and directories.

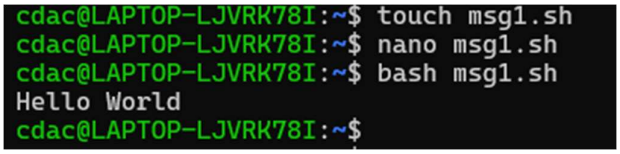
3. touch command is used to create a new file. mkdir command is used to create a new directory.

4.cat command is used to concatenate files.

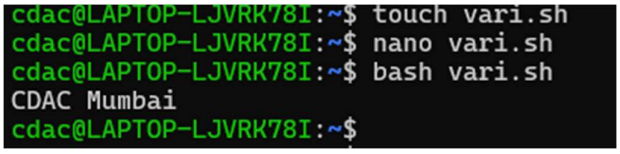
5. mv command is used to rename files when 2 files names are passed as arguments.

PART -4

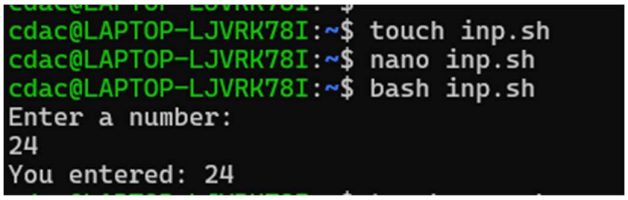
1.



2.



3.



4.

```
cdac@LAPTOP-LJVRK78I:~$ touch sum.sh
cdac@LAPTOP-LJVRK78I:~$ nano sum.sh
cdac@LAPTOP-LJVRK78I:~$ bash sum.sh
Enter first number:
12
Enter second number:
13
sum of 12 and 13 is: 25
```

5.

```
cdac@LAPTOP-LJVRK78I:~$ touch eo
cdac@LAPTOP-LJVRK78I:~$ touch eo.sh
cdac@LAPTOP-LJVRK78I:~$ nano eo.sh
cdac@LAPTOP-LJVRK78I:~$ bash eo.sh
Enter a number:
23
23 is odd.
```

```
cdac@LAPTOP-LJVRK78I:~$ bash eo.sh
Enter a number:
12
12 is even.
```

6.

```
cdac@LAPTOP-LJVRK78I:~$ touch whileloop.sh
cdac@LAPTOP-LJVRK78I:~$ nano whileloop.sh
cdac@LAPTOP-LJVRK78I:~$ bash whileloop.sh
1
2
3
4
5
```

7.

```
cdac@LAPTOP-LJVRK78I:~$ touch compareten
cdac@LAPTOP-LJVRK78I:~$ nano compareten
cdac@LAPTOP-LJVRK78I:~$ bash filee.sh
File exists
```

8.

```
cdac@LAPTOP-LJVRK78I:~$ touch compareten
cdac@LAPTOP-LJVRK78I:~$ bash compareten
Enter a number:
12
The number is greater than 10.
```

9.

```
cdac@LAPTOP-LJVRK78I:~$ touch table
cdac@LAPTOP-LJVRK78I:~$ nano table
cdac@LAPTOP-LJVRK78I:~$ nano table
cdac@LAPTOP-LJVRK78I:~$ bash table
   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@LAPTOP-LJVRK78I:~$
```

10.

```
cdac@LAPTOP-LJVRK78I:~$
cdac@LAPTOP-LJVRK78I:~$ touch square
cdac@LAPTOP-LJVRK78I:~$ nano square
cdac@LAPTOP-LJVRK78I:~$ bash square
Enter a number (enter a negative number to exit):
4
Square of 4 is: 16
Enter a number (enter a negative number to exit):
-4
Negative number entered. Exiting...
cdac@LAPTOP-LJVRK78I:~$ cat square

while true; do
    echo "Enter a number (enter a negative number to exit): "
    read number
    if [ $number -lt 0 ]; then
        echo "Negative number entered. Exiting..."
        break
    fi

    square=$((number * number))


    echo "Square of $number is: $square"
done
cdac@LAPTOP-LJVRK78I:~$
```

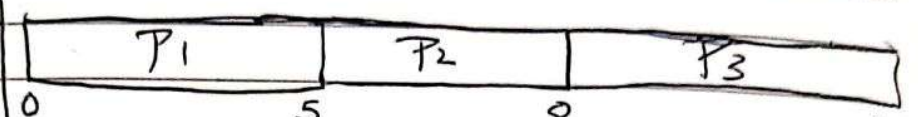
PART -5

+ C + J

DATE (CT - AT) (TAT - BT)

Q1	Arrival time	Burst time	TAT	WT
P1	0	5	5	0
P2	1	3	7	4
P3	2	6	12	6

RT | *CPU - A.T



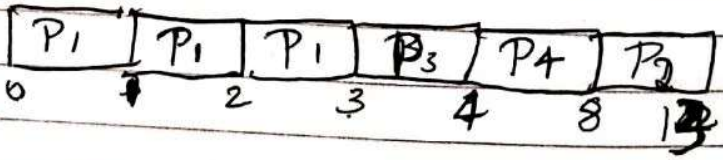
0
4
6

Av. TAT = $\frac{0+4+6}{3} = 3.33$

CT - AT

Q2 ⇒	AT	BT	TAT
P1	0	13 2	3
P2	1	5	13
P3	2	1	2
P4	3	4	5

SJF



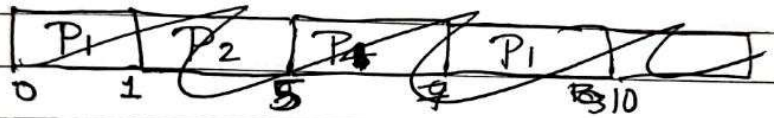
Av TAT = $3+13+6 = \frac{22}{4}$

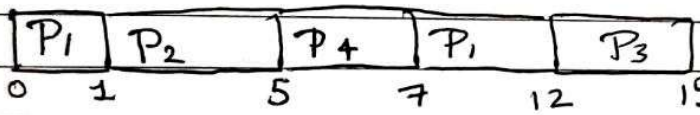
$= 5.5$

DATE

Q3 ⇒

	A.T	B.T	Priority	CT-AT TAT	TAT-BT WT
P_1	0	~~6~~ 5	3	12	6
P_2	1	4	1	4	0
P_3	2	7	4	17	10
P_4	3	2	2	4	2





$$\text{Av WT} = \frac{18}{4} = 4.5$$

Q4 ⇒

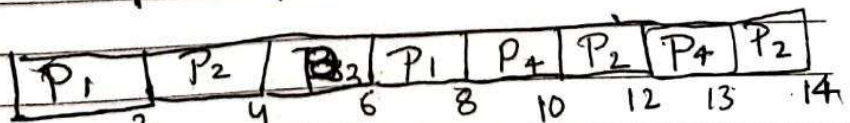
	A.T	B.T	TAT CT-AT
P_1	0	~~4~~ ~~2~~ 1	8
P_2	1	1 ~~5~~ ~~3~~	13
P_3	2	~~2~~ 4	4
P_4	3	~~3~~ 1	10

$\delta = 2$

Av. TAT

$= \frac{8+13+4+10}{4}$

$= \frac{35}{4} = 8.75$



R.Q