Assignment 2
Part A

1. **echo "Hello, World!"** – It Will print the Hello, World! On Screen

2. **name="Productive"** – it will create a variable name and will assign the value Productive

3. **touch file.txt** – it will create a file with .txt extension

4. **ls -a** – it will list all file and directories including hidden

5. **rm file.txt** – it will remove file

6. **cp file1.txt file2.txt** – it will copy the content of file1 into file2

7. **mv file.txt /path/to/directory/** – it will move the file on provided location

8. **chmod 755 script.sh** – it will change the permission of file.

9. **grep "pattern" file.txt** – it will search the text into file

10. **kill PID** – will terminate the process with provided PID

11. **mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt**
    =1st it will make a directory mydir then cd will change directory in that directory and it will create a file file.txt then echo will print text and it will save to file.txt and last it will show the content of file.txt on terminal.

12. **ls -l | grep ".txt"** – it will list file with .txt extension.

13. **cat file1.txt file2.txt | sort | uniq** – it display the combine content of both file with sorting it and also it will remove duplicate lines form content.

14. **ls -l | grep "^d"** – Lists only directories from the current location in long (-l) format.

15. **grep -r "pattern" /path/to/directory/** – it will search recursively for pattern word in all file for that location.

16. **cat file1.txt file2.txt | sort | uniq –d** – it will display the common duplicate lines

17. **chmod 644 file.txt** – it change the permission of for owner = read & write group = read other = read.

18. **cp -r source_directory destination_directory** – it copy whole content of source directory to destination directory.

19. **find /path/to/search -name "*.txt"** – search all file with .txt in location

20. **chmod u+x file.txt** – give permission of execute to owner.

21. **echo $PATH** – current PATH environment variable

Assignment 2

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. – **False**
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. – **True**

Identify the Incorrect Commands:

1. chmodx is used to change file permissions. - **Incorrect**
2. cpy is used to copy files and directories. - **Incorrect**
3. mkfile is used to create a new file. - **Incorrect**
4. catx is used to concatenate files. - **Incorrect**
5. rn is used to rename files. - **Incorrect**

Assignment 2

Part C

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

```
cdac@DESKTOP-M002MBJ:~$ vi asign1
cdac@DESKTOP-M002MBJ:~$ ./asign1
-bash: ./asign1: Permission denied
cdac@DESKTOP-M002MBJ:~$ chmod +x asign1
cdac@DESKTOP-M002MBJ:~$ ./asign
-bash: ./asign: No such file or directory
cdac@DESKTOP-M002MBJ:~$ ./asign1
Hello,World!
cdac@DESKTOP-M002MBJ:~$
```

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

cdac@DESKTOP-M002MBJ: ~

```
#!/bin/bash
name=" CDAC Mumbai"
echo $name
~
```

```
cdac@DESKTOP-M002MBJ:~$ vi asign1
cdac@DESKTOP-M002MBJ:~$ ./asign1
CDAC Mumbai
```

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

```
#!/bin/bash
echo "Heyy, Enter the Number : "
read num
echo $num
```

```
cdac@DESKTOP-M002MBJ:~$ ./asign1
Heyy, Enter the Number :
45
45
```

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

```
#!/bin/bash
echo "Additon"
echo "Enter NO 1 : "
read num1
echo "Enter No 2 : "
read num2
(( sum =  $num1 + $num2 ))
echo = $sum
```

```
cdac@DESKTOP-M002MBJ:~$ ./asign1
Additon
Enter NO 1 :
45
Enter No 2 :
10
= 55
```

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

```
#!/bin/bash
echo "Additon"
echo "Enter NO : "
read num
if [[ $num%2 -eq  0  ]] then
        echo "Its Even !"
else
        echo "Its Odd"
fi
~
```

```
cdac@DESKTOP-M002MBJ:~$ ./asign1
Additon
Enter NO :
66
Its Even !
cdac@DESKTOP-M002MBJ:~$
```

Assignment 2

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

```
#!/bin/bash
for var1 in 1 2 3 4 5
do
        echo $var1
done
~
```

```
cdac@DESKTOP-M002MBJ:~$ ./asign1
1
2
3
4
5
```

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

```
#!/bin/bash
i=1
while [ $i -le 5 ]
do
        echo $i
        ((i++))
done
```

```
cdac@DESKTOP-M002MBJ:~$ ./asign1
1
2
3
4
5
```

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".

```
#!/bin/bash
echo "Enter the file name :"
read file
if [ -f "$file" ]
then
        echo "File Available"
else
        echo "Not Available"
fi
```

```
cdac@DESKTOP-M002MBJ:~$ ./asign1
Enter the file name :
asign1
File Available
```

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.

```
#!/bin/bash
echo "Enter the Number :"
read num
if [ $num -ge 10 ]
then
        echo "Number is Greater than 10"
else
        echo "Number is Smaller then 10"
fi
```

```
cdac@DESKTOP-M002MBJ:~$ ./asign1
Enter the Number :
86
Number is Greater than 10
cdac@DESKTOP-M002MBJ:~$ ./asign1
Enter the Number :
5
Number is Smaller then 10
```

Assignment 2

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

```
#!/bin/bash
for i in {1..10}
do
        for j in {1..10}
        do
                echo -ne "$((i*j))\t"
        done
        echo " "
done
```

.

```
cdac@DESKTOP-M002MBJ:~$ ./asign1
1       2       3       4       5       6       7       8       9       10
2       4       6       8       10      12      14      16      18      20
3       6       9       12      15      18      21      24      27      30
4       8       12      16      20      24      28      32      36      40
5       10      15      20      25      30      35      40      45      50
6       12      18      24      30      36      42      48      54      60
7       14      21      28      35      42      49      56      63      70
8       16      24      32      40      48      56      64      72      80
9       18      27      36      45      54      63      72      81      90
10      20      30      40      50      60      70      80      90      100
```

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.

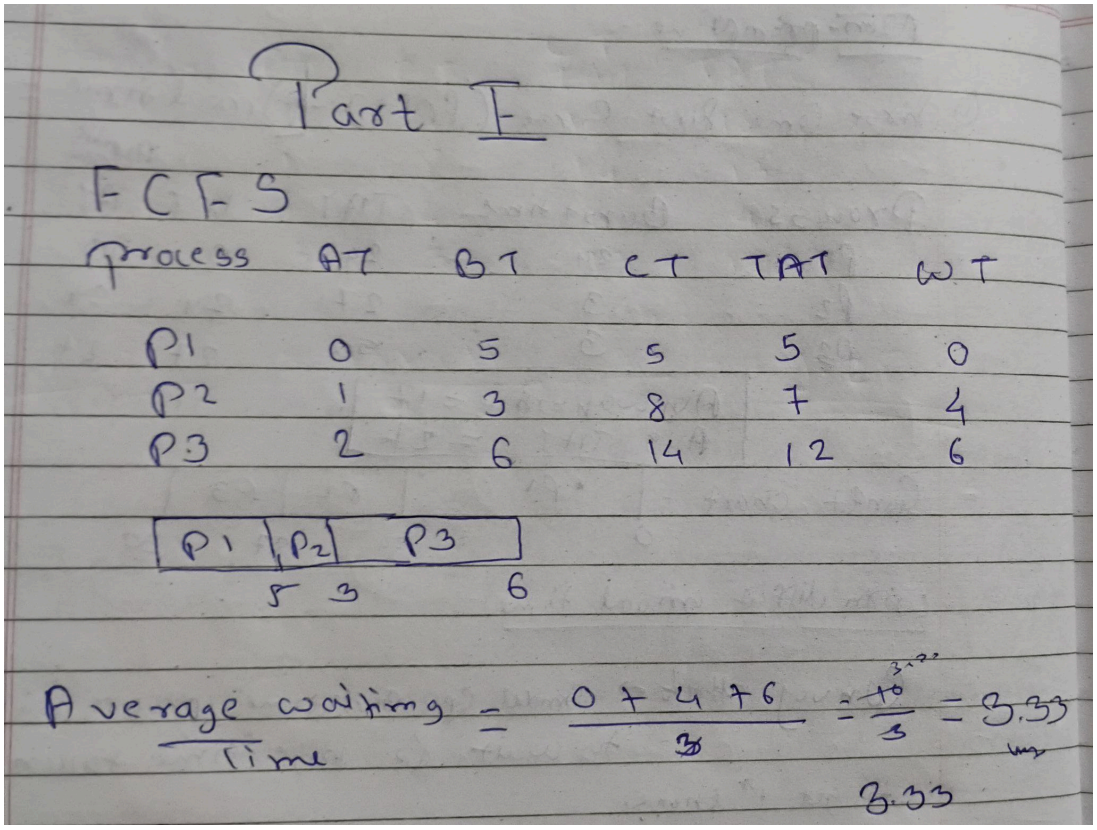
```
#!/bin/bash
while true
do
        echo "Enter the Number : "
        read no
        if [ $no -le  0 ]
        then
                echo "No is below 0"
                break
        else
                echo "Square of $no is  "$((no*no))

        fi
done
```

```
cdac@DESKTOP-M002MBJ:~$ ./asign1
Enter the Number :
2
Square of 2 is  4
Enter the Number :
8
Square of 8 is  64
Enter the Number :
6
Square of 6 is  36
Enter the Number :
74235
Square of 74235 is  5510835225
Enter the Number :
-8
No is below 0
```

Assignment 2

Part E



Part I

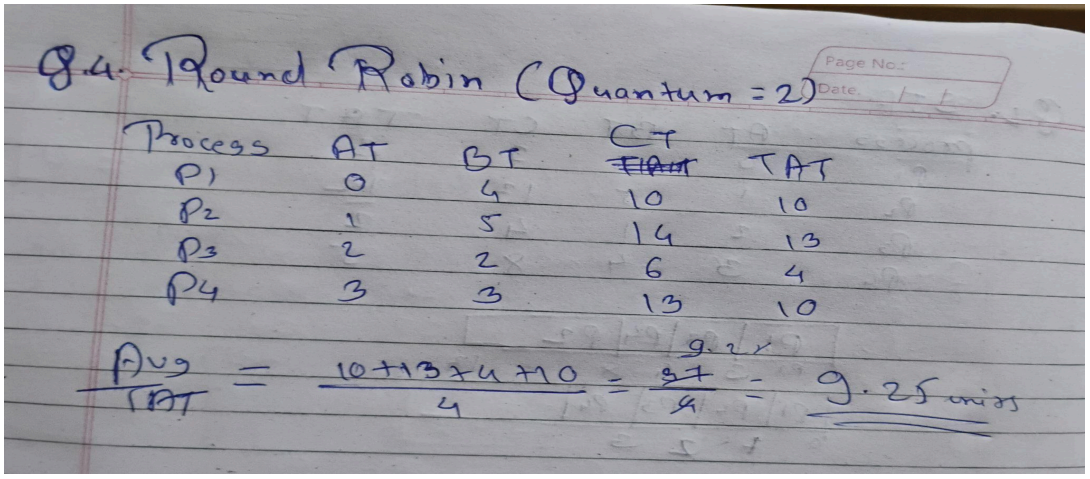
FCFS

Process	AT	BT	CT	TAT	WT
P1	0	5	5	5	0
P2	1	3	8	7	4
P3	2	6	14	12	6

P1	P2	P3
5	3	6

Average waiting time = $\frac{0+4+6}{3} = \frac{10}{3} = 3.33$ ms

3.33



Q.4. Round Robin (Quantum = 2)

Process	AT	BT	CT	TAT
P1	0	4	10	10
P2	1	5	14	13
P3	2	2	6	4
P4	3	3	13	10

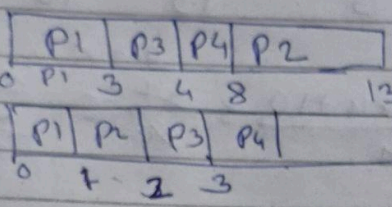
Avg TAT = $\frac{10+13+4+10}{4} = \frac{37}{4} = 9.25$ ms

9.25

Assignment 2

Q.2. SJF

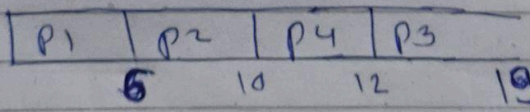
Process	AT	BT	CT	TAT
P1	0	3	3	3
P2	1	5	13	12
P3	2	1	4	2
P4	3	4	8	5

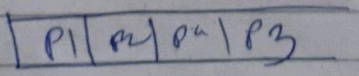


TAT AVG = $\frac{3+12+2+5}{4} = \frac{22}{4}$ = 5.5 units

Q.3. Priority Scheduling (Non preemptive)

Process	AT	BT	Priority	CT	TAT	WT
P1	0	6	3	6	6	0
P2	1	4	1	10	9	5
P3	2	7	4	19	17	10
P4	3	2	2	12	9	7





AVG TAT = $\frac{0+5+10+7}{4} = \frac{22}{4}$ 5.5

5.5 units