Name: Harshada Nande

Concepts of Operating System
Assignment 2
Part A

What will the following commands do?

- **echo "Hello, World!"**
  - The command ‘echo “Hello, World”’ will display the text “Hello, World” in the terminal. echo command used to print the Hello world on the screen.

- **name="Productive"**
  - The command ‘name=”Productive”’ it will used assign the value to the variable.
  - ‘name’ is the shell variable name and ‘Productive’ is the value assign to the name variable.

- **touch file.txt**
  - The command ‘touch file.txt’ is used to create the empty file named ‘file.txt’ . ‘touch ‘ command is used to create the empty file.

- **ls -a**
  - the “ls -a” command is used to list the all files and directories and also hidden files and directories in the current directory.
  - “ls” is use for list the file and directory.

- **rm file.txt**
  - The command ‘rm file.txt’ is used to delete the file.txt from the current directory .
  - rm command is used for deleted the file from directory.

- **cp file1.txt file2.txt**
  - The command “cp file1.txt file2.txt” it will be used for the copy the file1.txt data into the file2.txt.
  - cp command use for the copy the data from one file to another file.

- **mv file.txt /path/to/directory/**
  - **‘**mv’ command is used for move the file from current location to the specific directory.

- File.txt is the file it is mv from specific directory.

- **chmod 755 script.sh**
- This command is used to assign read, write and execute permission to owner, group and other users.
- In the ‘chmod 755 script.sh’ command gives the read , write and execute permission to the owner and read and execute permission to the group and other users for ‘script.sh’ file.

- **grep "pattern" file.txt**
- The ‘grep’ command is used for searching the specific word in the file. ‘grep "pattern" file.txt’ in this command ‘pattern’ is a word and grep is used for searching the “pattern” word in the file.txt file.
- And display all line in file.txt that contains the ‘pattern’ word

- **kill PID**

- This command will terminate the process whose PID is mentioned in the command.
- Since the above command doesn’t contain any process id, above command will result in an error.

- **ls -l | grep ".txt"**
- This command uses piping to combine the output of both ls and grep command. ls -l is used to display the contents of current directory with details and grep “.txt” command is used to display all the files conating .txt pattern in their name
- **ls -l | grep "^d"**
- ls command lists the files and directories in long format. grep "^d" command filters the output to show only lines that start with "d" which in the ls -l output indicates directories

- **grep -r "pattern" /path/to/directory/**
- The ‘grep’ command is used for searching the specific word in the file. ‘grep -r "pattern" /path/to/directory/ ’ this command is use for the searching the “pattern” word in the all the file in current directory and subdirectories.

- **chmod 644 file.txt**
  - This command is used to assign read, write and execute permission to owner, group and other users.
  - In the 'chmod 644 file.txt' command gives the read and write permission to the owner and only read permission to the group and other users for 'file.txt' file.

- **cp -r source_directory destination_directory**
  - The above command is used to copy the source_directory to destination directory. This is done by using -r option so that all files in source_directory are copied recursively**.**
- **find /path/to/search -name "*.txt"**
  - find command is used for searching the files and directories. Given command searches /path/to/search directory and its subdirectories for any file ending with .txt pattern**.**

- **chmod u+x file.txt**
  - This command is used to gives execute permissions for file.txt file to the user of the file.

- **echo $PATH**
  - This command used for display the value of system environment variable that stores directories where executable program are located.

Part B

Identify True or False:

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

Ans: Ture

2. mv is used to move files and directories.

Ans: True

3. cd is used to copy files and directories.

Ans: False

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

Ans: True

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

Ans: True

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

Ans: True

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

Ans: True

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

Ans: True

Identify the Incorrect Commands:

1. chmodx is used to change file permissions.

Ans: Correct command is ‘chmod’.

2. cpy is used to copy files and directories.

Ans: Correct command is “cp”

3. mkfile is used to create a new file.

Ans: Correct command is “touch”

4. catx is used to concatenate files.

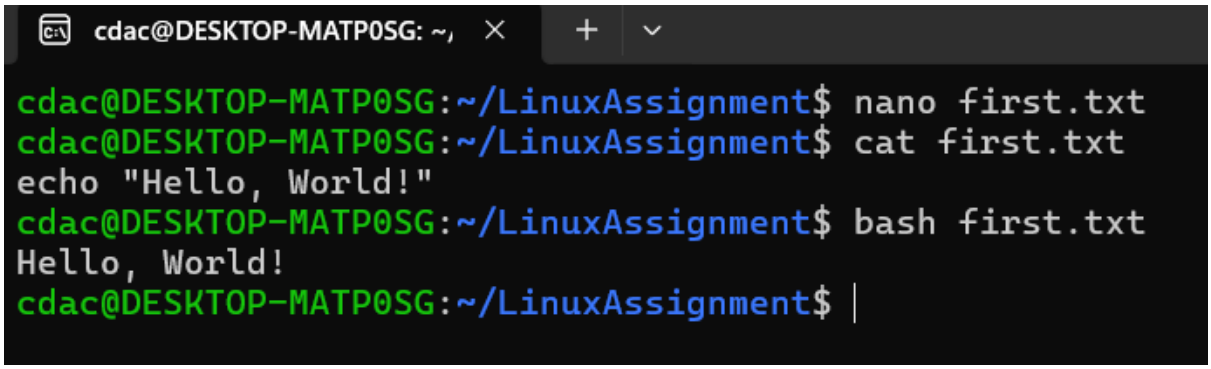
Ans: Correct command is “cat”

5. rn is used to rename files.

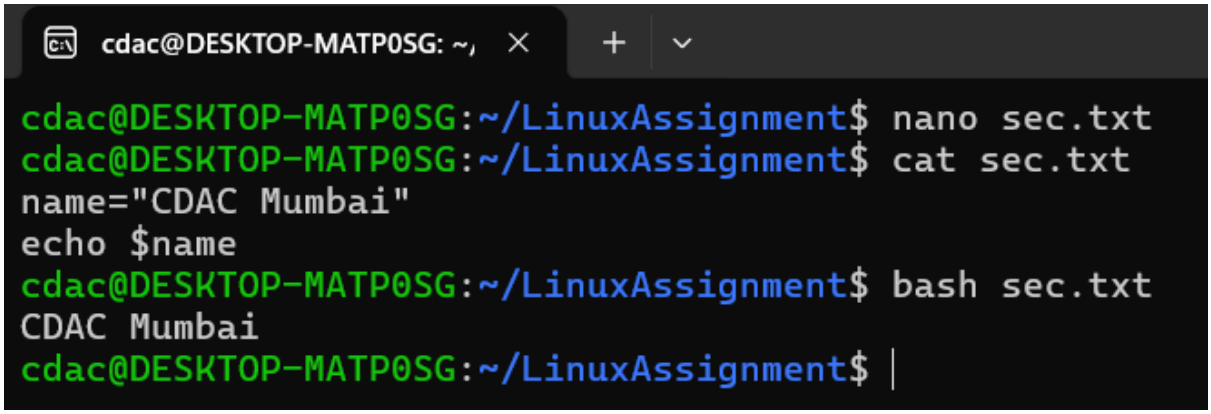
Ans: Correct command is “mv”

Part C

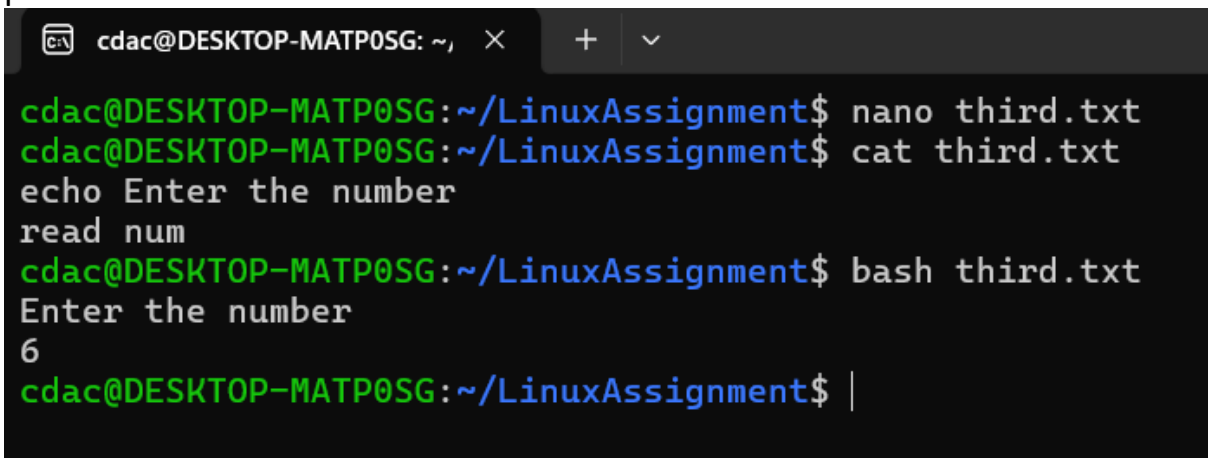
Question 1: Write a shell script that prints "Hello, World!" to the terminal.



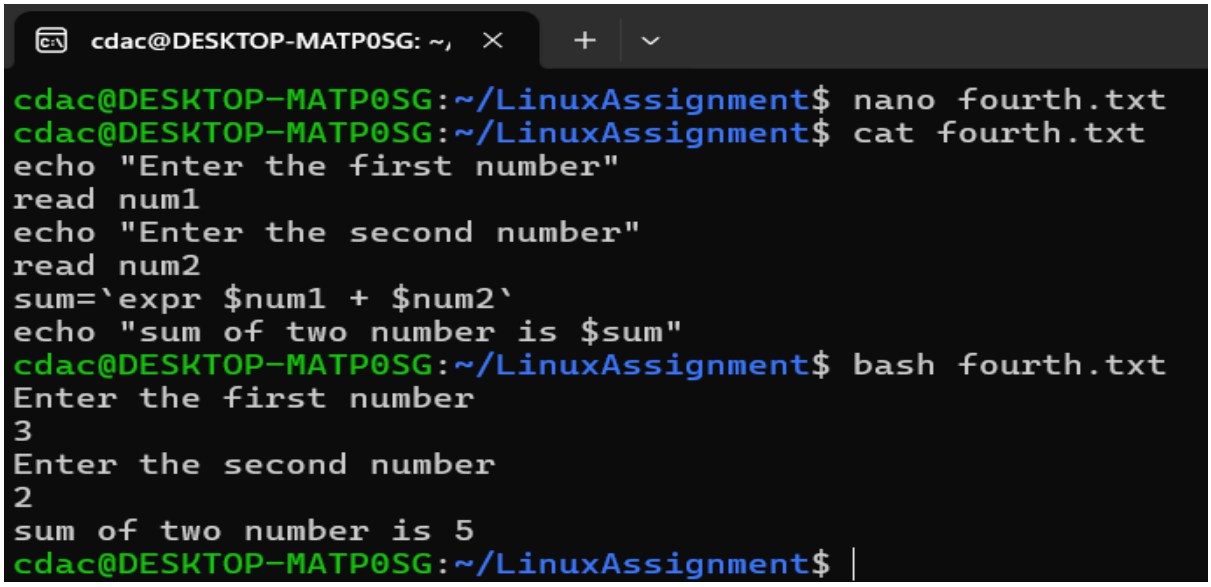
Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.



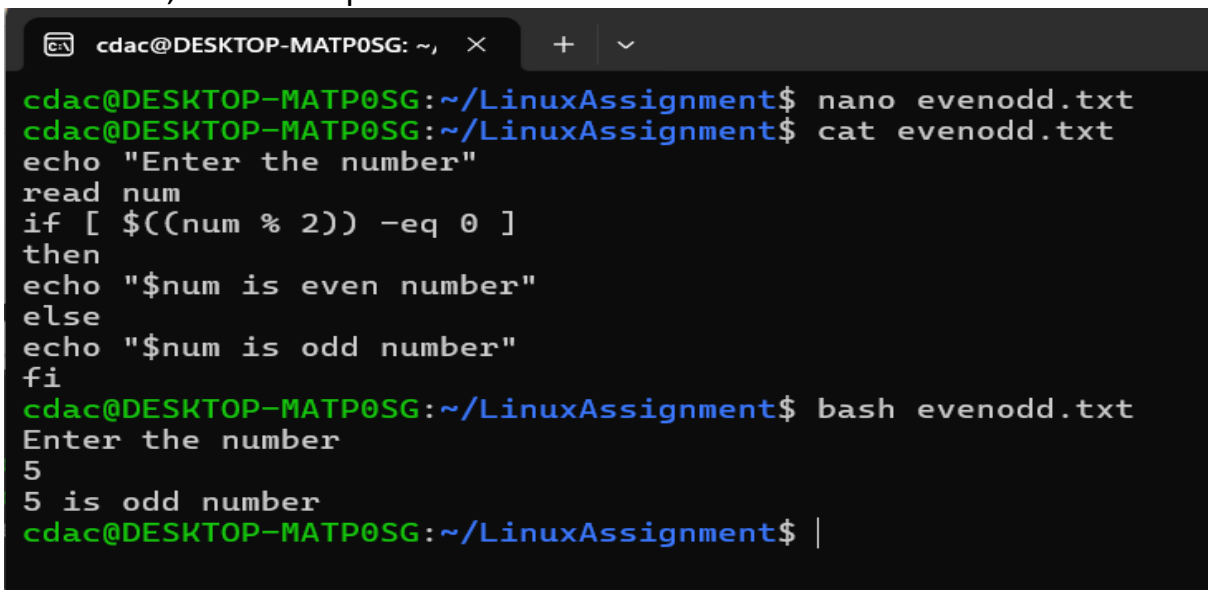
Question 3: Write a shell script that takes a number as input from the user and prints it.



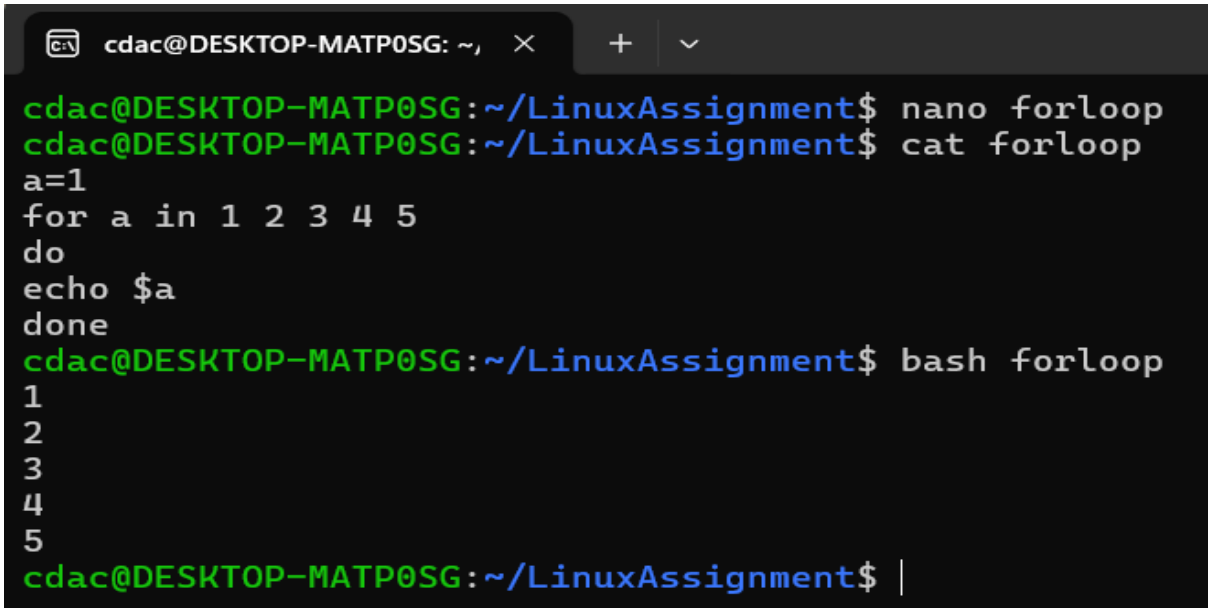
Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.



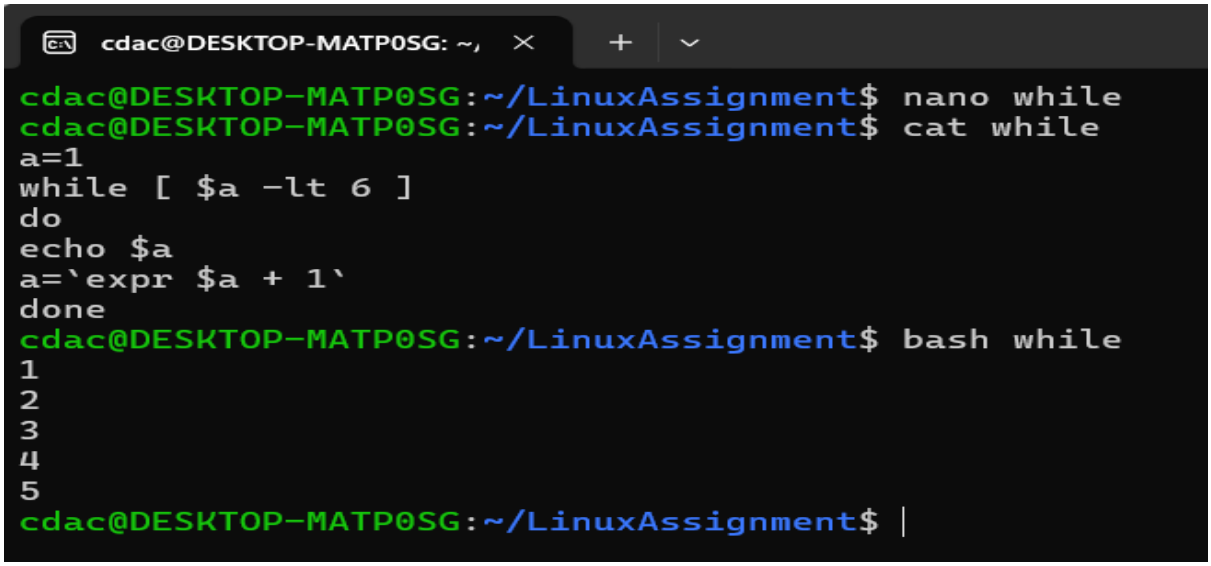
Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".



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



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

```
cdac@DESKTOP-MATP0SG: ~,
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ nano exist
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ cat exist
echo "enter the name"
read name
if [ -e $name ]
then
echo "File is exist"
else
echo "file is not exist"
fi
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ bash exist
enter the name
while
File is exist
cdac@DESKTOP-MATP0SG:~/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.

```
cdac@DESKTOP-MATP0SG: ~,
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ nano if
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ cat if
echo Enter the number
read num
if [ $num -gt 10 ]
then
echo "Number is greater than 10"
else
echo "number is less thsn 10"
fi
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ bash if
Enter the number
11
Number is greater than 10
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ bash if
Enter the number
3
number is less thsn 10
cdac@DESKTOP-MATP0SG:~/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.

```
cdac@DESKTOP-MATP0SG: ~,
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ cat for1
for i in 1 2 3 4 5
do
for j in 1 2 3 4 5
do
echo "$i * $j = " $((i * j))
done
echo
done
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ bash for1
1 * 1 =  1
1 * 2 =  2
1 * 3 =  3
1 * 4 =  4
1 * 5 =  5

2 * 1 =  2
2 * 2 =  4
2 * 3 =  6
2 * 4 =  8
2 * 5 =  10

3 * 1 =  3
3 * 2 =  6
3 * 3 =  9
3 * 4 =  12
3 * 5 =  15

4 * 1 =  4
4 * 2 =  8
4 * 3 =  12
4 * 4 =  16
4 * 5 =  20

5 * 1 =  5
5 * 2 =  10
5 * 3 =  15
5 * 4 =  20
5 * 5 =  25

cdac@DESKTOP-MATP0SG:~/LinuxAssignment$
```

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.

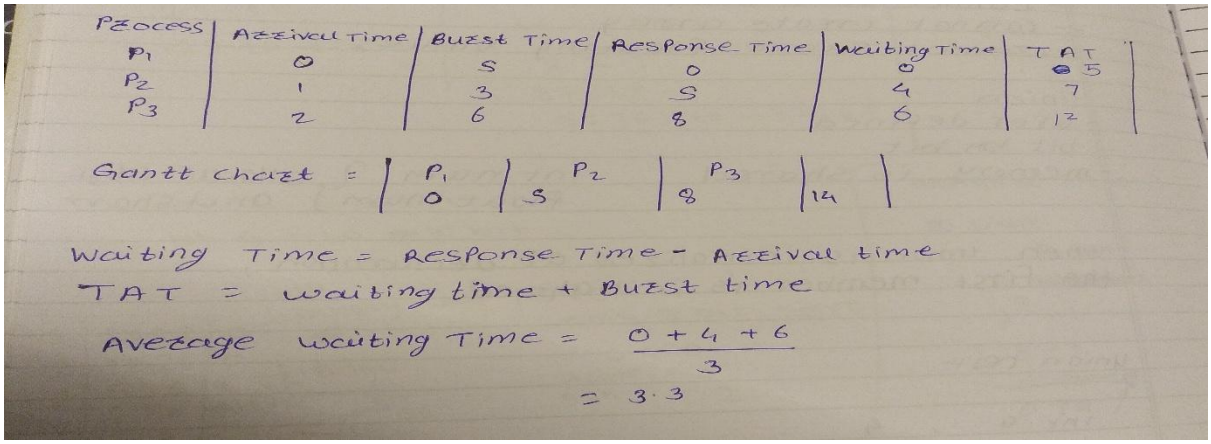
```
cdac@DESKTOP-MATP0SG: ~,
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ nano while1
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ cat while1
while true
do
echo "Enter the number"
read no

if [ $no -lt 0 ]
then
echo "entering negative number "
break

fi

square=$((no * no))
echo "The square of number is $square"
done
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$ bash while1
Enter the number
2
The square of number is 4
Enter the number
4
The square of number is 16
Enter the number
-6
entering negative number
cdac@DESKTOP-MATP0SG:~/LinuxAssignment$
```

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.



Process	Arrival Time	Burst Time	Response Time	Waiting Time	TAT
P_1	0	5	0	0	5
P_2	1	3	5	4	7
P_3	2	6	8	6	12

Gantt chart :

P_1	P_2	P_3	
0	5	8	14

Waiting Time = Response Time - Arrival time

TAT = waiting time + Burst time

Average waiting Time = $\frac{0 + 4 + 6}{3}$

$= 3.3$

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

| Process | Arrival Time | Burst Time |

|---------|--------------|------------|

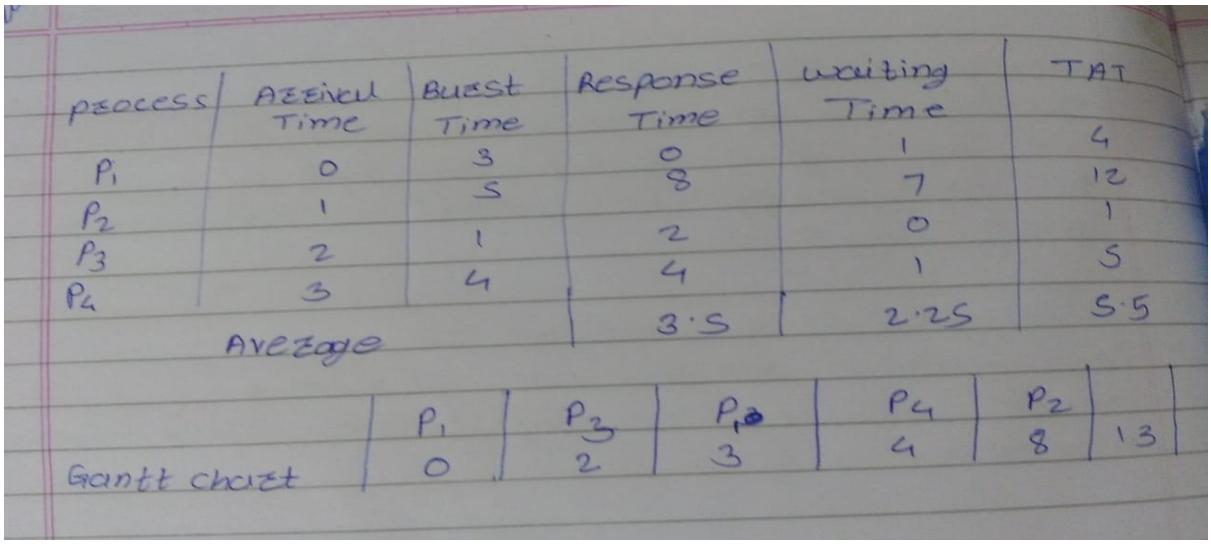
| P1 | 0 | 3 |

| P2 | 1 | 5 |

| P3 | 2 | 1 |

| P4 | 3 | 4 |

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



process	Arrival Time	Burst Time	Response Time	waiting Time	TAT
P_1	0	3	0	1	4
P_2	1	5	8	7	12
P_3	2	1	2	0	1
P_4	3	4	4	1	5
Average			3.5	2.25	5.5

Gantt chart

P_1	P_3	P_4	P_2	
0	2	3	8	13