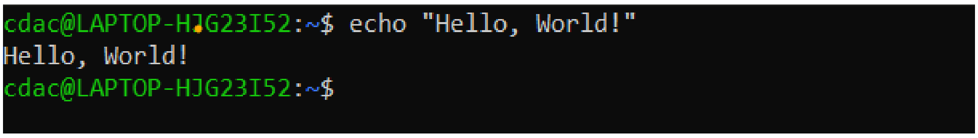
Assignment no.2

Concepts of Operating system

Part A

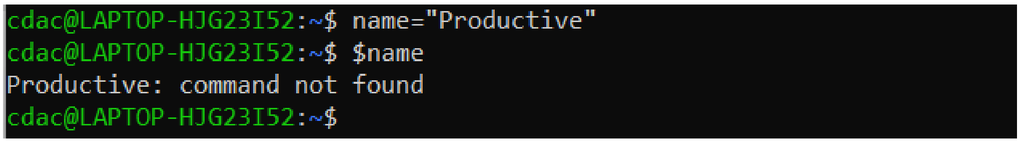
What will the following commands do?

1. echo "Hello, World!"
   ans: echo is the keyword that means print , so this command will display Hello, World! On the console.



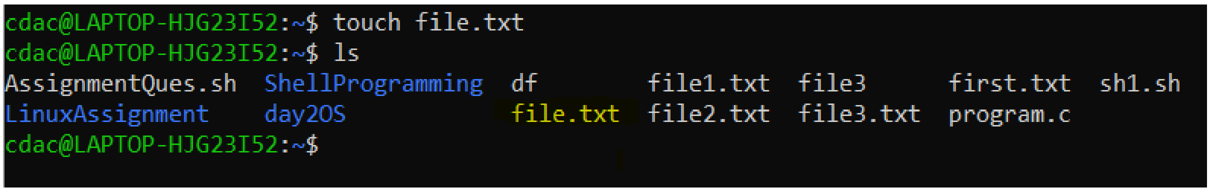
```
cdac@LAPTOP-HJG23I52:~$ echo "Hello, World!"
Hello, World!
cdac@LAPTOP-HJG23I52:~$
```

2. name="Productive"
   ans: In this command the terminal saves the name variable until the system is dead.



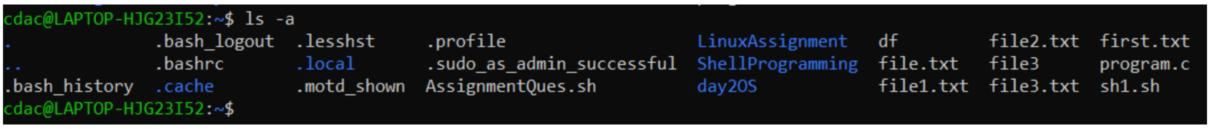
```
cdac@LAPTOP-HJG23I52:~$ name="Productive"
cdac@LAPTOP-HJG23I52:~$ $name
Productive: command not found
cdac@LAPTOP-HJG23I52:~$
```

3. touch file.txt
   ans: 'touch' command is used to create a file so this will create a file named file.txt in the directory.



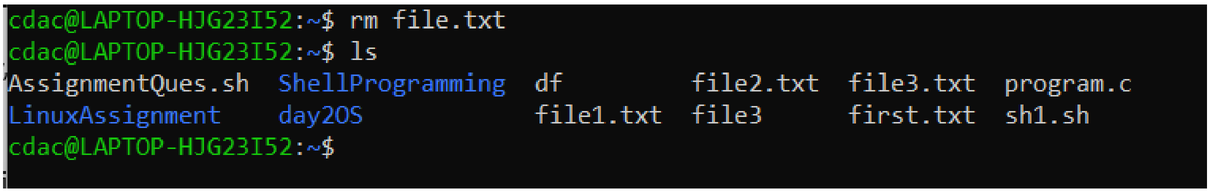
```
cdac@LAPTOP-HJG23I52:~$ touch file.txt
cdac@LAPTOP-HJG23I52:~$ ls
AssignmentQues.sh  ShellProgramming  df        file1.txt  file3      first.txt  sh1.sh
LinuxAssignment    day2OS            file.txt  file2.txt  file3.txt  program.c
cdac@LAPTOP-HJG23I52:~$
```

4. ls -a
   ans: 'ls' command is used to display all the files in the directory. '-a' is a flag which is used to show all the hidden files and folders in the directory.



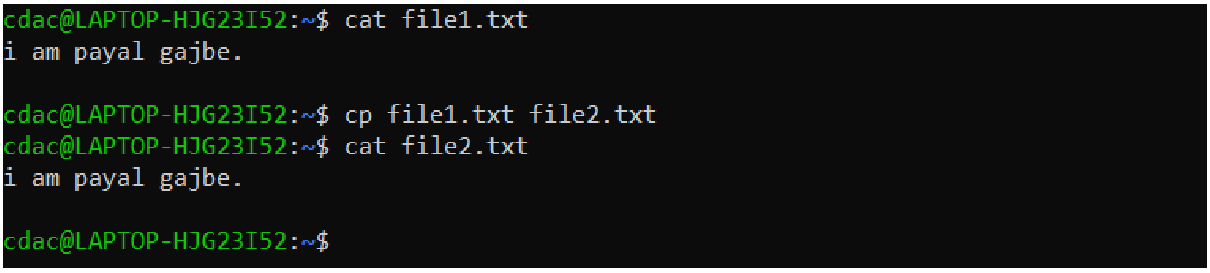
```
cdac@LAPTOP-HJG23I52:~$ ls -a
.              .bash_logout  .lesshst     .profile                   LinuxAssignment   df         file2.txt  first.txt
..             .bashrc       .local       .sudo_as_admin_successful  ShellProgramming  file.txt   file3      program.c
.bash_history  .cache        .motd_shown  AssignmentQues.sh          day2OS            file1.txt  file3.txt  sh1.sh
cdac@LAPTOP-HJG23I52:~$
```

5. rm file.txt
   ans: 'rm' is the command used to remove a file from the directory, so in this the file.txt is removed from the directory.



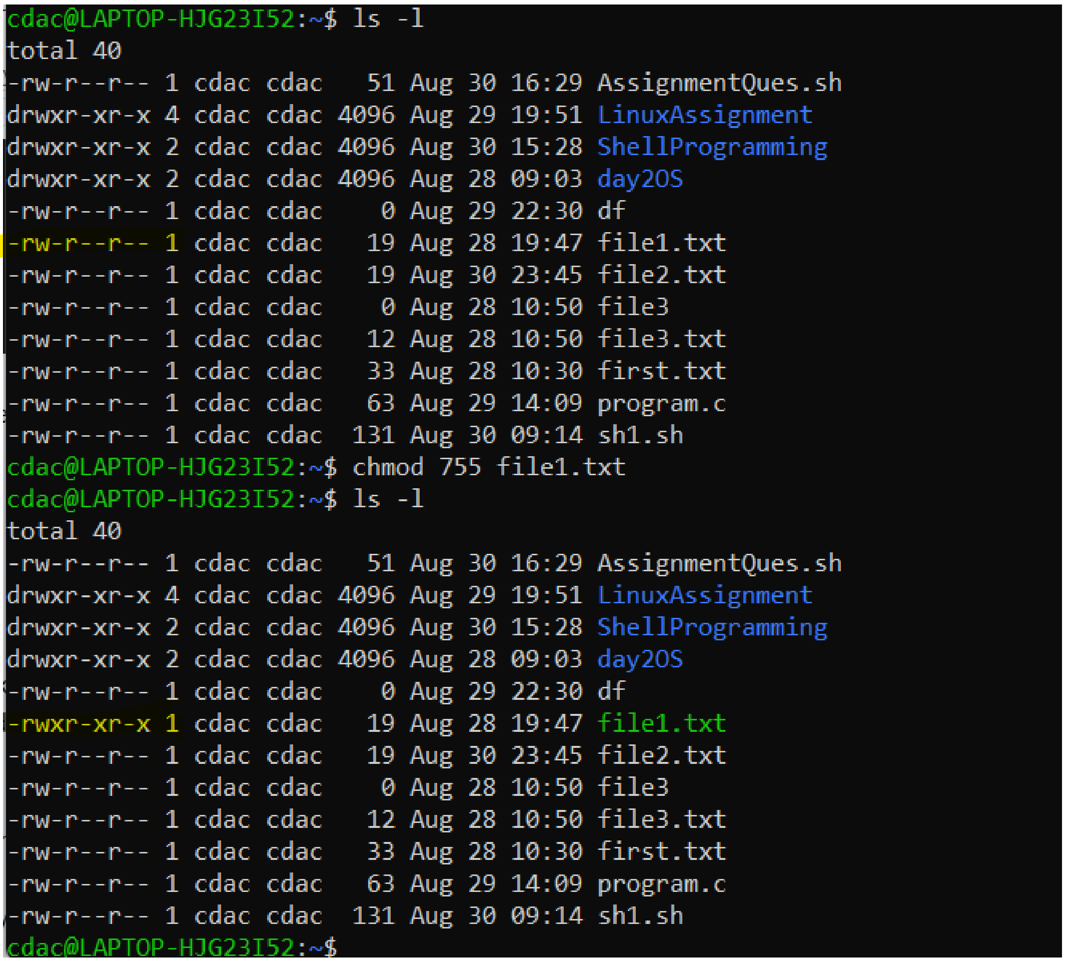
```
cdac@LAPTOP-HJG23I52:~$ rm file.txt
cdac@LAPTOP-HJG23I52:~$ ls
AssignmentQues.sh  ShellProgramming  df         file2.txt  file3.txt  program.c
LinuxAssignment    day2OS            file1.txt  file3      first.txt  sh1.sh
cdac@LAPTOP-HJG23I52:~$
```

6. cp file1.txt file2.txt
   ans: 'cp' command is used to copy content of one file to another file. The syntax is cp <sourcefile> <targetfile> so, the contents of file1.txt will get copied in file2.txt.



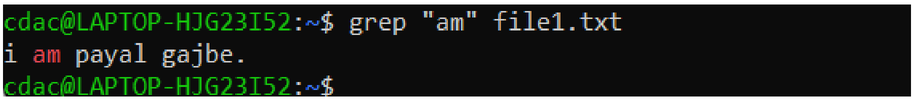
```
cdac@LAPTOP-HJG23I52:~$ cat file1.txt
i am payal gajbe.

cdac@LAPTOP-HJG23I52:~$ cp file1.txt file2.txt
cdac@LAPTOP-HJG23I52:~$ cat file2.txt
i am payal gajbe.

cdac@LAPTOP-HJG23I52:~$
```

7. mv file.txt /path/to/directory/
   ans: : 'mv' command is used to move content of one file to another file. The syntax is mv <sourcefile> <targetfile> so, the contents of file.txt will get moved in the given directory path.
8. chmod 755 script.sh
   ans: Is used to give permissions to the scipt.sh file.



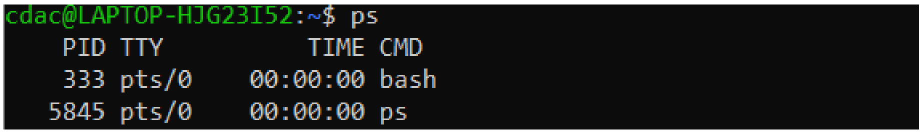
```
cdac@LAPTOP-HJG23I52:~$ ls -l
total 40
-rw-r--r-- 1 cdac cdac   51 Aug 30 16:29 AssignmentQues.sh
drwxr-xr-x 4 cdac cdac 4096 Aug 29 19:51 LinuxAssignment
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:28 ShellProgramming
drwxr-xr-x 2 cdac cdac 4096 Aug 28 09:03 day2OS
-rw-r--r-- 1 cdac cdac    0 Aug 29 22:30 df
-rw-r--r-- 1 cdac cdac   19 Aug 28 19:47 file1.txt
-rw-r--r-- 1 cdac cdac   19 Aug 30 23:45 file2.txt
-rw-r--r-- 1 cdac cdac    0 Aug 28 10:50 file3
-rw-r--r-- 1 cdac cdac   12 Aug 28 10:50 file3.txt
-rw-r--r-- 1 cdac cdac   33 Aug 28 10:30 first.txt
-rw-r--r-- 1 cdac cdac   63 Aug 29 14:09 program.c
-rw-r--r-- 1 cdac cdac  131 Aug 30 09:14 sh1.sh
cdac@LAPTOP-HJG23I52:~$ chmod 755 file1.txt
cdac@LAPTOP-HJG23I52:~$ ls -l
total 40
-rw-r--r-- 1 cdac cdac   51 Aug 30 16:29 AssignmentQues.sh
drwxr-xr-x 4 cdac cdac 4096 Aug 29 19:51 LinuxAssignment
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:28 ShellProgramming
drwxr-xr-x 2 cdac cdac 4096 Aug 28 09:03 day2OS
-rw-r--r-- 1 cdac cdac    0 Aug 29 22:30 df
-rwxr-xr-x 1 cdac cdac   19 Aug 28 19:47 file1.txt
-rw-r--r-- 1 cdac cdac   19 Aug 30 23:45 file2.txt
-rw-r--r-- 1 cdac cdac    0 Aug 28 10:50 file3
-rw-r--r-- 1 cdac cdac   12 Aug 28 10:50 file3.txt
-rw-r--r-- 1 cdac cdac   33 Aug 28 10:30 first.txt
-rw-r--r-- 1 cdac cdac   63 Aug 29 14:09 program.c
-rw-r--r-- 1 cdac cdac  131 Aug 30 09:14 sh1.sh
cdac@LAPTOP-HJG23I52:~$
```

9. grep "pattern" file.txt



```
cdac@LAPTOP-HJG23I52:~$ grep "am" file1.txt
i am payal gajbe.
cdac@LAPTOP-HJG23I52:~$
```

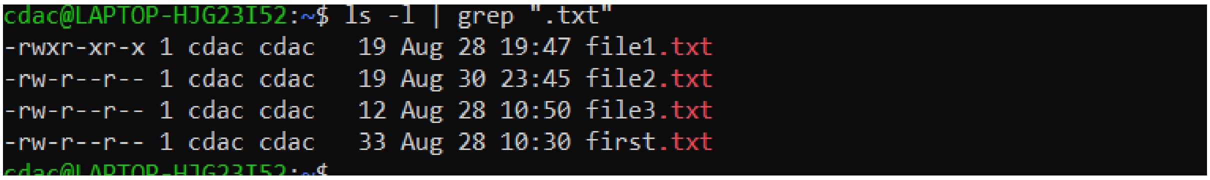
10. kill PID
    ans: 'kill' command is used to kill a process and in this case it will kill PID with process ID.



```
cdac@LAPTOP-HJG23I52:~$ ps
    PID TTY          TIME CMD
    333 pts/0    00:00:00 bash
   5845 pts/0    00:00:00 ps
```

11. mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt

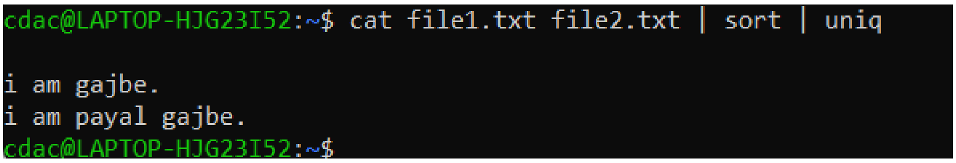
12. ls -l | grep ".txt"



```
cdac@LAPTOP-HJG23I52:~$ ls -l | grep ".txt"
-rwxr-xr-x 1 cdac cdac   19 Aug 28 19:47 file1.txt
-rw-r--r-- 1 cdac cdac   19 Aug 30 23:45 file2.txt
-rw-r--r-- 1 cdac cdac   12 Aug 28 10:50 file3.txt
-rw-r--r-- 1 cdac cdac   33 Aug 28 10:30 first.txt
cdac@LAPTOP-HJG23I52:~$
```

13. cat file1.txt file2.txt | sort | uniq

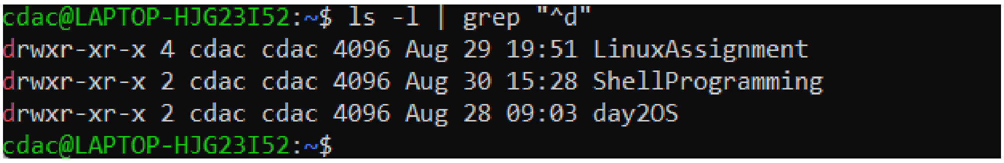
ans: This concatenate file1 and file2 and then sort it and display unique values in it.



```
cdac@LAPTOP-HJG23I52:~$ cat file1.txt file2.txt | sort | uniq

i am gajbe.
i am payal gajbe.
cdac@LAPTOP-HJG23I52:~$
```

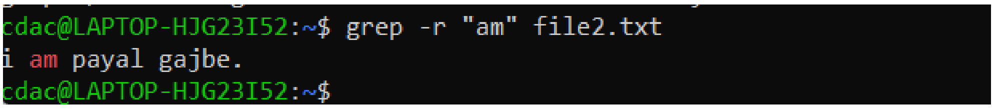
14. ls -l | grep "^d"

Ans: This gives the list of all the directories in the current directory with details(long form).



```
cdac@LAPTOP-HJG23I52:~$ ls -l | grep "^d"
drwxr-xr-x 4 cdac cdac 4096 Aug 29 19:51 LinuxAssignment
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:28 ShellProgramming
drwxr-xr-x 2 cdac cdac 4096 Aug 28 09:03 day2OS
cdac@LAPTOP-HJG23I52:~$
```

15. grep -r "pattern" /path/to/directory/



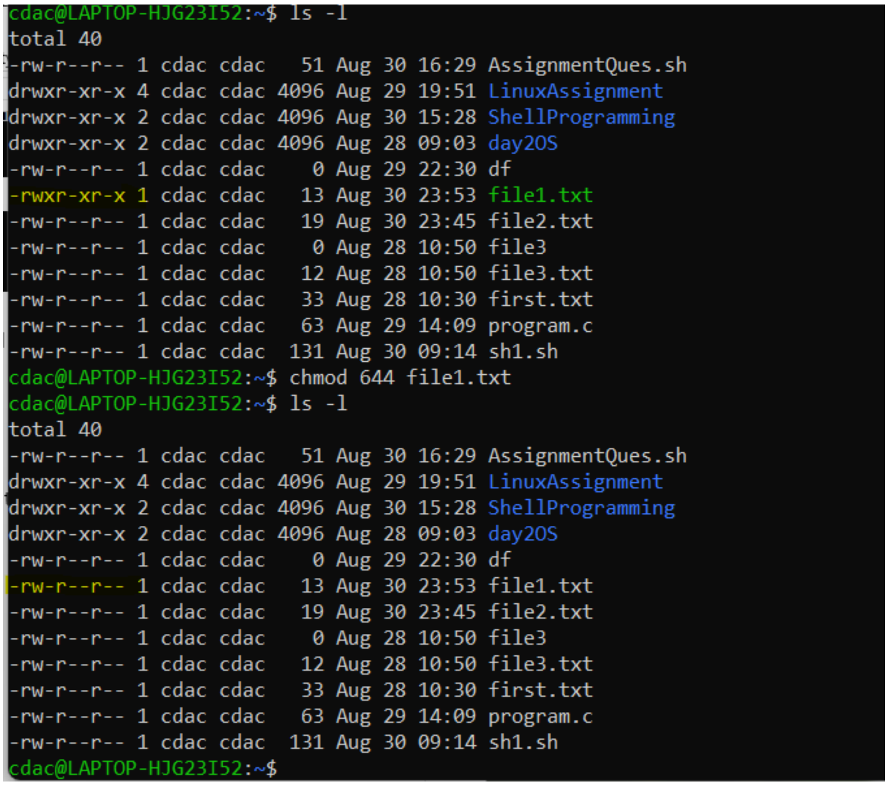
```
cdac@LAPTOP-HJG23I52:~$ grep -r "am" file2.txt
i am payal gajbe.
cdac@LAPTOP-HJG23I52:~$
```

16. cat file1.txt file2.txt | sort | uniq –d

ans: This command will concate file1 and file2 then sort them and then give unique values in them.

17. chmod 644 file.txt

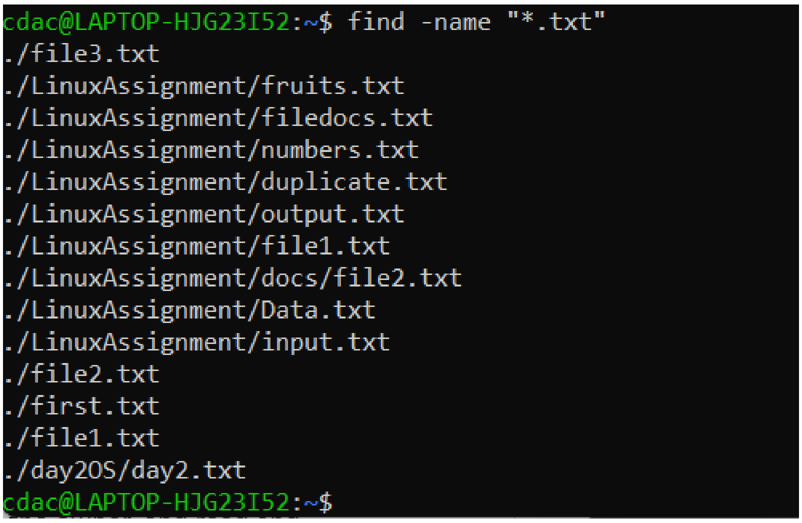
Ans: used to remove permissions from the file.txt.



```
cdac@LAPTOP-HJG23I52:~$ ls -l
total 40
-rw-r--r-- 1 cdac cdac   51 Aug 30 16:29 AssignmentQues.sh
drwxr-xr-x 4 cdac cdac 4096 Aug 29 19:51 LinuxAssignment
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:28 ShellProgramming
drwxr-xr-x 2 cdac cdac 4096 Aug 28 09:03 day2OS
-rw-r--r-- 1 cdac cdac    0 Aug 29 22:30 df
-rwxr-xr-x 1 cdac cdac   13 Aug 30 23:53 file1.txt
-rw-r--r-- 1 cdac cdac   19 Aug 30 23:45 file2.txt
-rw-r--r-- 1 cdac cdac    0 Aug 28 10:50 file3
-rw-r--r-- 1 cdac cdac   12 Aug 28 10:50 file3.txt
-rw-r--r-- 1 cdac cdac   33 Aug 28 10:30 first.txt
-rw-r--r-- 1 cdac cdac   63 Aug 29 14:09 program.c
-rw-r--r-- 1 cdac cdac  131 Aug 30 09:14 sh1.sh
cdac@LAPTOP-HJG23I52:~$ chmod 644 file1.txt
cdac@LAPTOP-HJG23I52:~$ ls -l
total 40
-rw-r--r-- 1 cdac cdac   51 Aug 30 16:29 AssignmentQues.sh
drwxr-xr-x 4 cdac cdac 4096 Aug 29 19:51 LinuxAssignment
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:28 ShellProgramming
drwxr-xr-x 2 cdac cdac 4096 Aug 28 09:03 day2OS
-rw-r--r-- 1 cdac cdac    0 Aug 29 22:30 df
-rw-r--r-- 1 cdac cdac   13 Aug 30 23:53 file1.txt
-rw-r--r-- 1 cdac cdac   19 Aug 30 23:45 file2.txt
-rw-r--r-- 1 cdac cdac    0 Aug 28 10:50 file3
-rw-r--r-- 1 cdac cdac   12 Aug 28 10:50 file3.txt
-rw-r--r-- 1 cdac cdac   33 Aug 28 10:30 first.txt
-rw-r--r-- 1 cdac cdac   63 Aug 29 14:09 program.c
-rw-r--r-- 1 cdac cdac  131 Aug 30 09:14 sh1.sh
cdac@LAPTOP-HJG23I52:~$
```

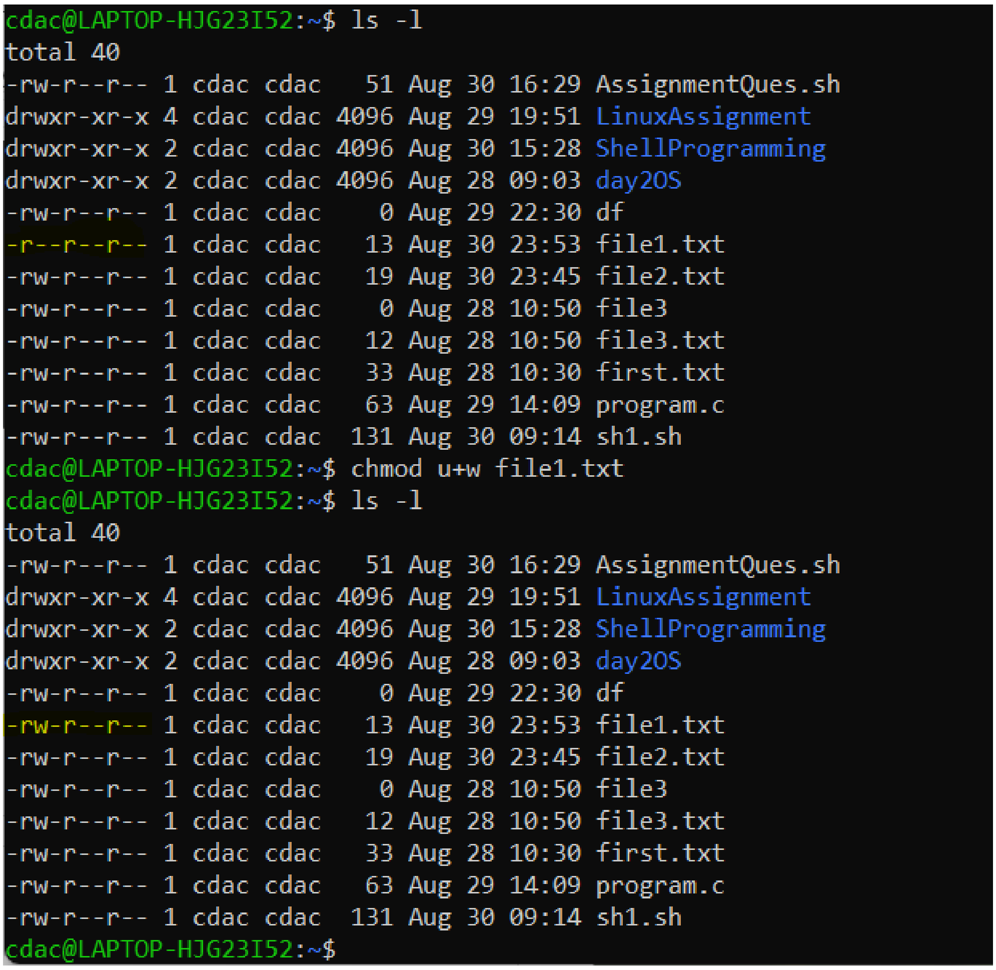
18. cp -r source_directory destination_directory
    Ans: ‘cp -r’ is used to copy file recursive means r is used to search through all the files through that directory.

19. find /path/to/search -name "*.txt"
    Ans: this is used to find files with txt extension.



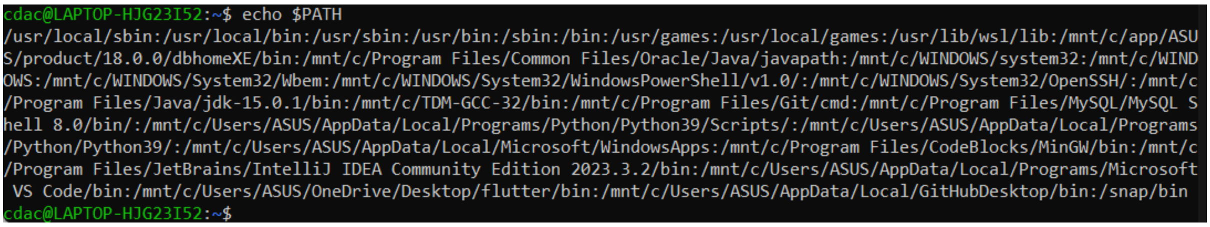
```
cdac@LAPTOP-HJG23I52:~$ find -name "*.txt"
./file3.txt
./LinuxAssignment/fruits.txt
./LinuxAssignment/filedocs.txt
./LinuxAssignment/numbers.txt
./LinuxAssignment/duplicate.txt
./LinuxAssignment/output.txt
./LinuxAssignment/file1.txt
./LinuxAssignment/docs/file2.txt
./LinuxAssignment/Data.txt
./LinuxAssignment/input.txt
./file2.txt
./first.txt
./file1.txt
./day20S/day2.txt
cdac@LAPTOP-HJG23I52:~$
```

20. chmod u+x file.txt
    ans: used to give execution permission to user.



```
cdac@LAPTOP-HJG23I52:~$ ls -l
total 40
-rw-r--r-- 1 cdac cdac   51 Aug 30 16:29 AssignmentQues.sh
drwxr-xr-x 4 cdac cdac 4096 Aug 29 19:51 LinuxAssignment
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:28 ShellProgramming
drwxr-xr-x 2 cdac cdac 4096 Aug 28 09:03 day20S
-rw-r--r-- 1 cdac cdac    0 Aug 29 22:30 df
-r--r--r-- 1 cdac cdac   13 Aug 30 23:53 file1.txt
-rw-r--r-- 1 cdac cdac   19 Aug 30 23:45 file2.txt
-rw-r--r-- 1 cdac cdac    0 Aug 28 10:50 file3
-rw-r--r-- 1 cdac cdac   12 Aug 28 10:50 file3.txt
-rw-r--r-- 1 cdac cdac   33 Aug 28 10:30 first.txt
-rw-r--r-- 1 cdac cdac   63 Aug 29 14:09 program.c
-rw-r--r-- 1 cdac cdac  131 Aug 30 09:14 sh1.sh
cdac@LAPTOP-HJG23I52:~$ chmod u+w file1.txt
cdac@LAPTOP-HJG23I52:~$ ls -l
total 40
-rw-r--r-- 1 cdac cdac   51 Aug 30 16:29 AssignmentQues.sh
drwxr-xr-x 4 cdac cdac 4096 Aug 29 19:51 LinuxAssignment
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:28 ShellProgramming
drwxr-xr-x 2 cdac cdac 4096 Aug 28 09:03 day20S
-rw-r--r-- 1 cdac cdac    0 Aug 29 22:30 df
-rw-r--r-- 1 cdac cdac   13 Aug 30 23:53 file1.txt
-rw-r--r-- 1 cdac cdac   19 Aug 30 23:45 file2.txt
-rw-r--r-- 1 cdac cdac    0 Aug 28 10:50 file3
-rw-r--r-- 1 cdac cdac   12 Aug 28 10:50 file3.txt
-rw-r--r-- 1 cdac cdac   33 Aug 28 10:30 first.txt
-rw-r--r-- 1 cdac cdac   63 Aug 29 14:09 program.c
-rw-r--r-- 1 cdac cdac  131 Aug 30 09:14 sh1.sh
cdac@LAPTOP-HJG23I52:~$
```

21. echo $PATH
    ans: This command diplays all the path of the system.



```
cdac@LAPTOP-HJG23I52:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/wsl/lib:/mnt/c/app/ASU
S/product/18.0.0/dbhomeXE/bin:/mnt/c/Program Files/Common Files/Oracle/Java/javapath:/mnt/c/WINDOWS/system32:/mnt/c/WIND
OWS:/mnt/c/WINDOWS/System32/Wbem:/mnt/c/WINDOWS/System32/WindowsPowerShell/v1.0/:/mnt/c/WINDOWS/System32/OpenSSH/:/mnt/c
/Program Files/Java/jdk-15.0.1/bin:/mnt/c/TDM-GCC-32/bin:/mnt/c/Program Files/Git/cmd:/mnt/c/Program Files/MySQL/MySQL S
hell 8.0/bin/:/mnt/c/Users/ASUS/AppData/Local/Programs/Python/Python39/Scripts/:/mnt/c/Users/ASUS/AppData/Local/Programs
/Python/Python39/:/mnt/c/Users/ASUS/AppData/Local/Microsoft/WindowsApps:/mnt/c/Program Files/CodeBlocks/MinGW/bin:/mnt/c
/Program Files/JetBrains/IntelliJ IDEA Community Edition 2023.3.2/bin:/mnt/c/Users/ASUS/AppData/Local/Programs/Microsoft
 VS Code/bin:/mnt/c/Users/ASUS/OneDrive/Desktop/flutter/bin:/mnt/c/Users/ASUS/AppData/Local/GitHubDesktop/bin:/snap/bin
cdac@LAPTOP-HJG23I52:~$
```

Part B

Identify True or False:

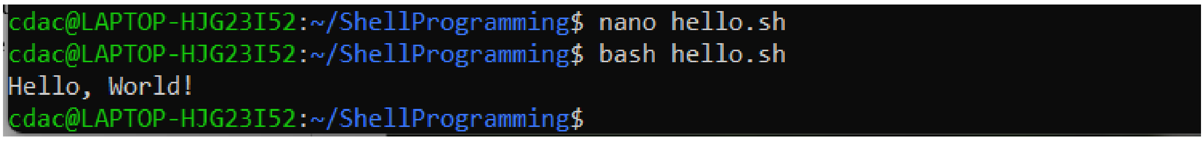
1. ls is used to list files and directories in a directory. **True**
2. mv is used to move files and directories. **True**
3. cd is used to copy files and directories. **True**
4. pwd stands for "print working directory" and displays the current directory. **True**
5. grep is used to search for patterns 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.
   Ans: ‘chmod’ is used to change the file permissions.
2. cpy is used to copy files and directories.
   Ans: ‘cp’ command is used to copy files and directories.
3. mkfile is used to create a new file.
   Ans: ‘mkdir’ command is used to create a new directory while ‘touch’, ’nano’ &’cat’ can be used to create a file.
4. catx is used to concatenate files.
   Ans: ‘cat’ is used to concatenate files and also can be used to display a file.
5. rn is used to rename files.
   Ans: ‘mv’ command is used to rename the files.

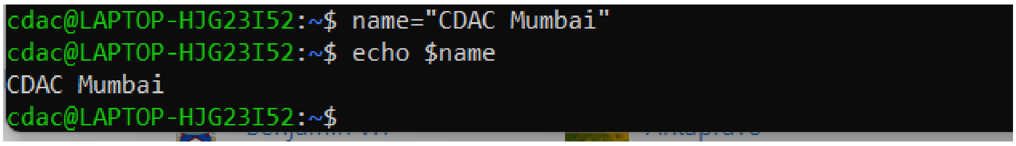
Part C

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



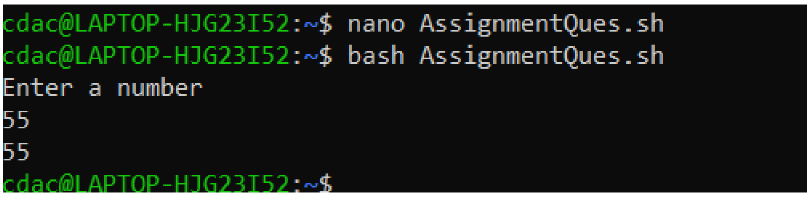
```
cdac@LAPTOP-HJG23I52:~/ShellProgramming$ nano hello.sh
cdac@LAPTOP-HJG23I52:~/ShellProgramming$ bash hello.sh
Hello, World!
cdac@LAPTOP-HJG23I52:~/ShellProgramming$
```

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



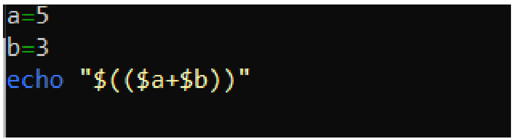
```
cdac@LAPTOP-HJG23I52:~$ name="CDAC Mumbai"
cdac@LAPTOP-HJG23I52:~$ echo $name
CDAC Mumbai
cdac@LAPTOP-HJG23I52:~$
```

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

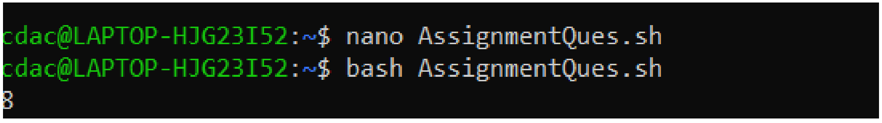


```
cdac@LAPTOP-HJG23I52:~$ nano AssignmentQues.sh
cdac@LAPTOP-HJG23I52:~$ bash AssignmentQues.sh
Enter a number
55
55
cdac@LAPTOP-HJG23I52:~$
```

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

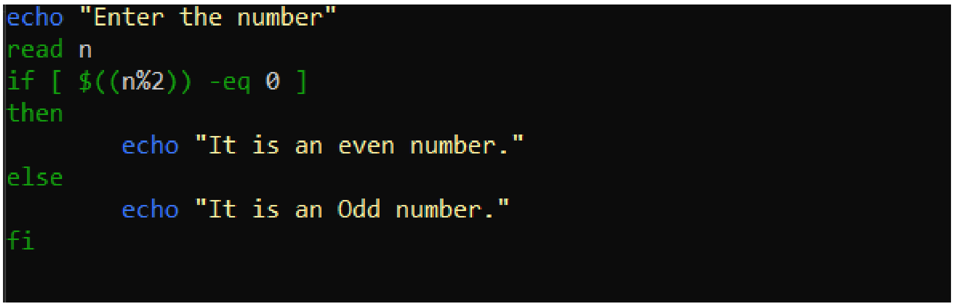


```
a=5
b=3
echo "$(($a+$b))"
```

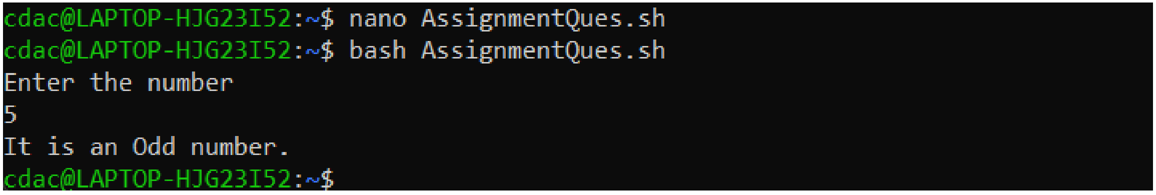


```
cdac@LAPTOP-HJG23I52:~$ nano AssignmentQues.sh
cdac@LAPTOP-HJG23I52:~$ bash AssignmentQues.sh
8
```

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

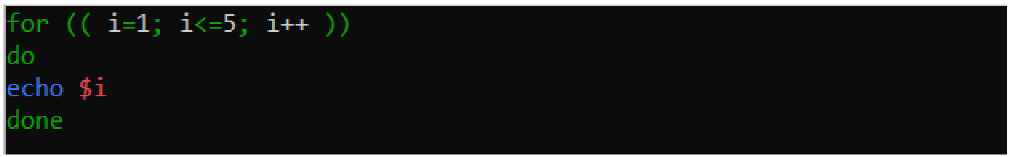


```
echo "Enter the number"
read n
if [ $((n%2)) -eq 0 ]
then
        echo "It is an even number."
else
        echo "It is an Odd number."
fi
```

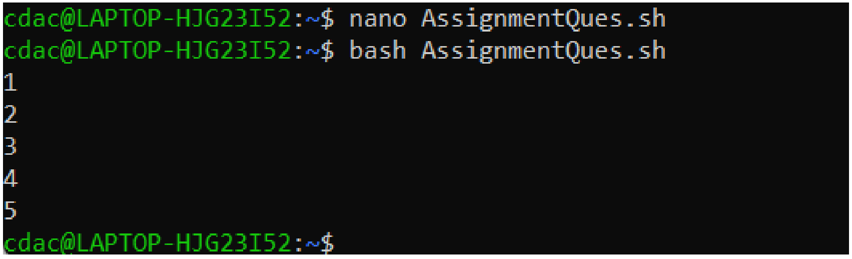


```
cdac@LAPTOP-HJG23I52:~$ nano AssignmentQues.sh
cdac@LAPTOP-HJG23I52:~$ bash AssignmentQues.sh
Enter the number
5
It is an Odd number.
cdac@LAPTOP-HJG23I52:~$
```

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

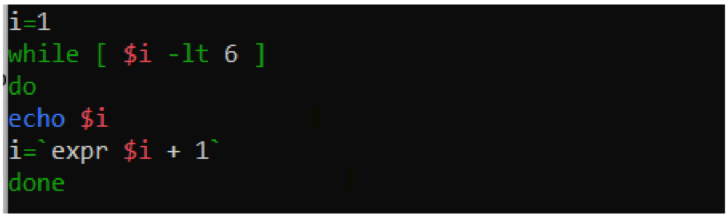


```
for (( i=1; i<=5; i++ ))
do
echo $i
done
```

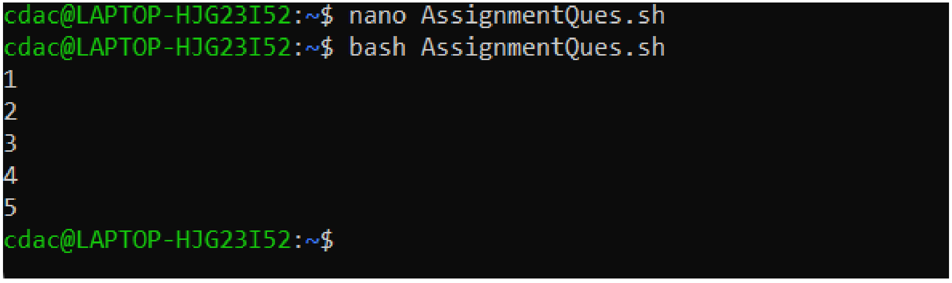


```
cdac@LAPTOP-HJG23I52:~$ nano AssignmentQues.sh
cdac@LAPTOP-HJG23I52:~$ bash AssignmentQues.sh
1
2
3
4
5
cdac@LAPTOP-HJG23I52:~$
```

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

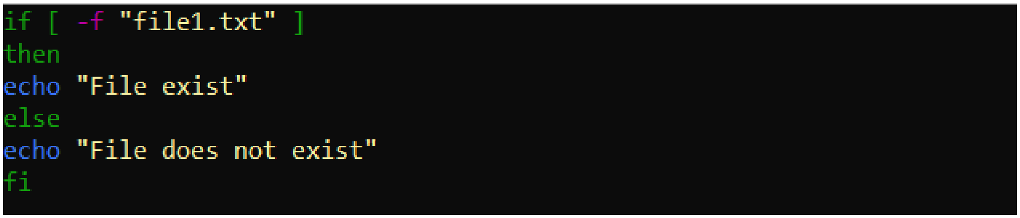


```
i=1
while [ $i -lt 6 ]
do
echo $i
i=`expr $i + 1`
done
```

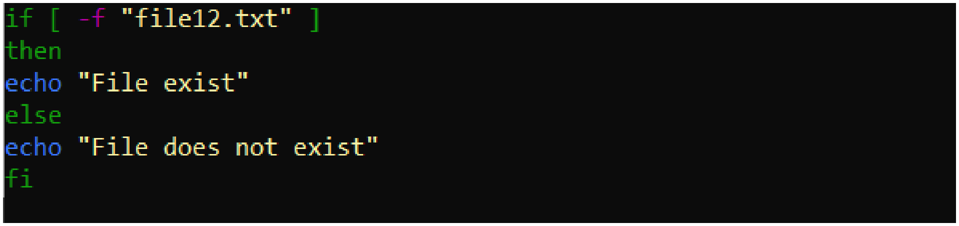


```
cdac@LAPTOP-HJG23I52:~$ nano AssignmentQues.sh
cdac@LAPTOP-HJG23I52:~$ bash AssignmentQues.sh
1
2
3
4
5
cdac@LAPTOP-HJG23I52:~$
```

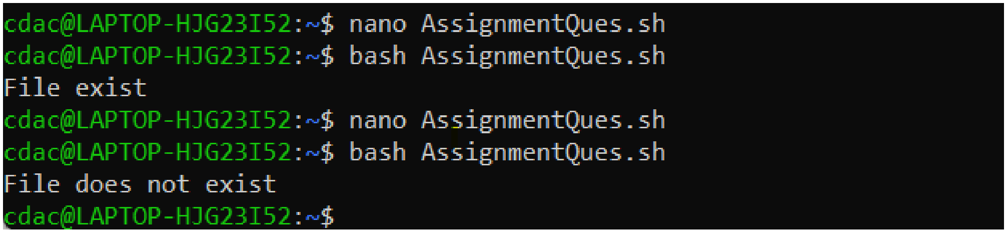
Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".



```
if [ -f "file1.txt" ]
then
echo "File exist"
else
echo "File does not exist"
fi
```

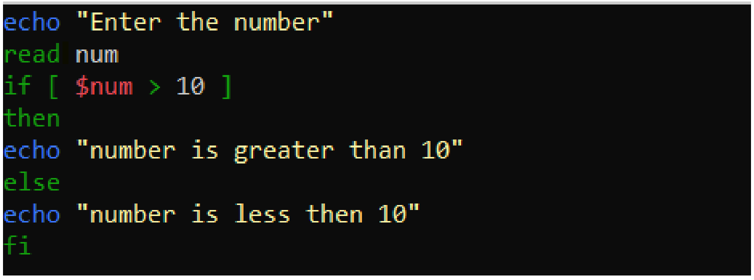


```
if [ -f "file12.txt" ]
then
echo "File exist"
else
echo "File does not exist"
fi
```

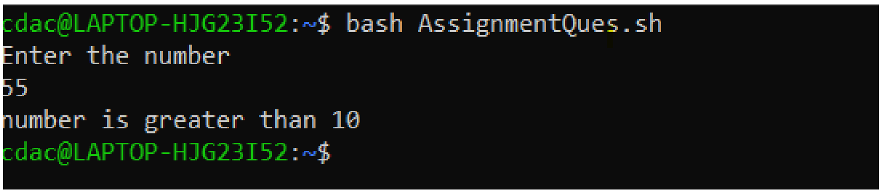


```
cdac@LAPTOP-HJG23I52:~$ nano AssignmentQues.sh
cdac@LAPTOP-HJG23I52:~$ bash AssignmentQues.sh
File exist
cdac@LAPTOP-HJG23I52:~$ nano AssignmentQues.sh
cdac@LAPTOP-HJG23I52:~$ bash AssignmentQues.sh
File does not exist
cdac@LAPTOP-HJG23I52:~$
```

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.

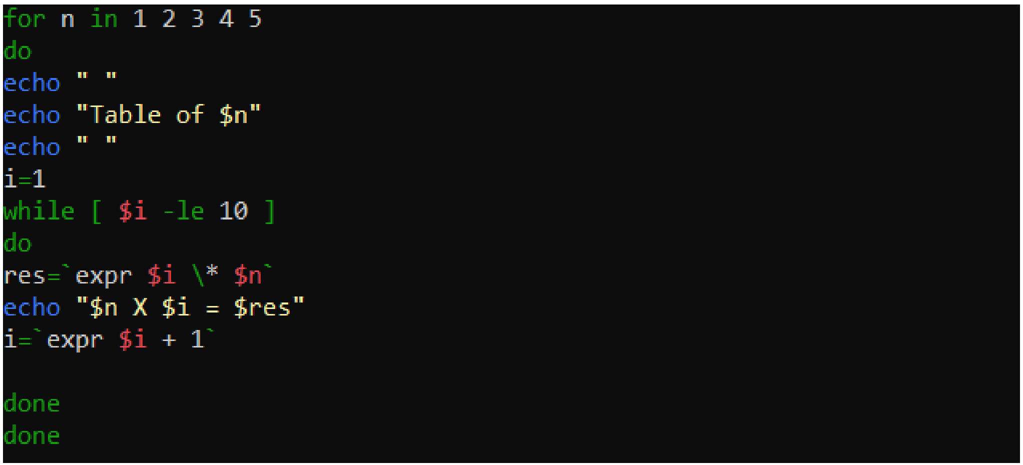


```
echo "Enter the number"
read num
if [ $num > 10 ]
then
echo "number is greater than 10"
else
echo "number is less then 10"
fi
```

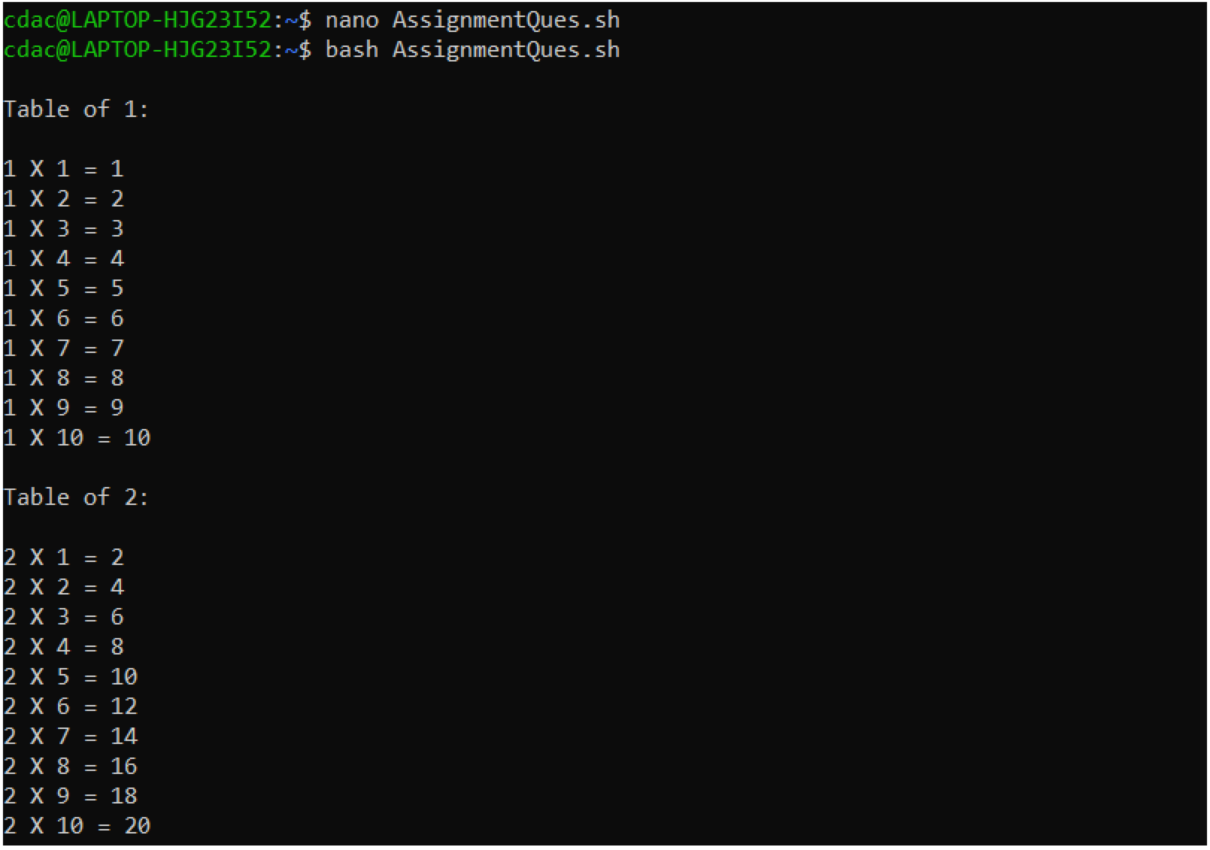


```
cdac@LAPTOP-HJG23I52:~$ bash AssignmentQues.sh
Enter the number
55
number is greater than 10
cdac@LAPTOP-HJG23I52:~$
```

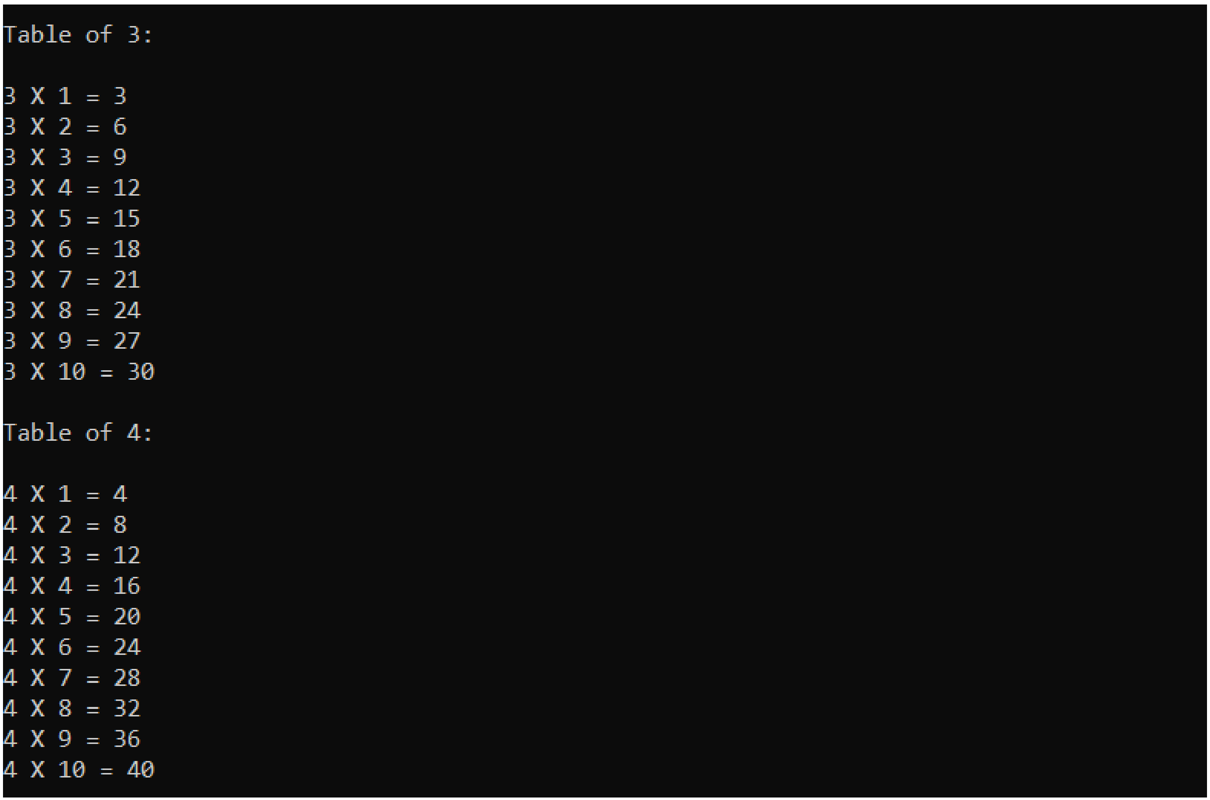
Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.



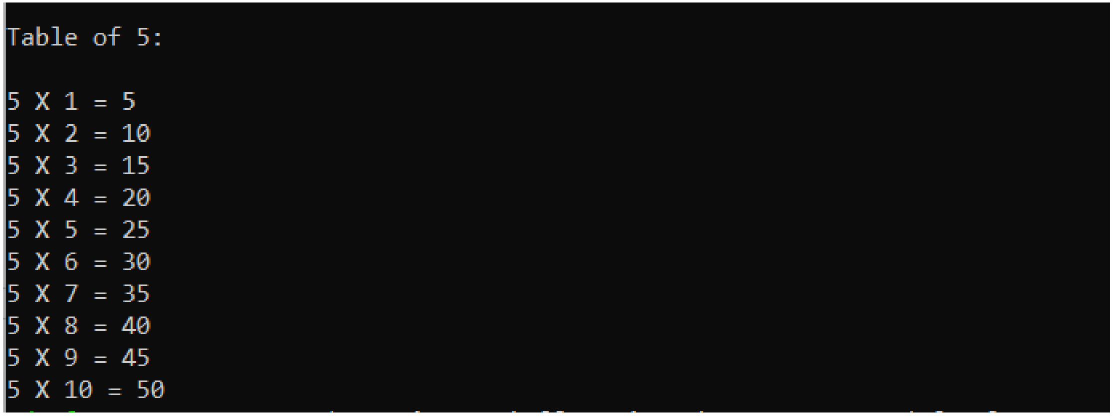
```
for n in 1 2 3 4 5
do
echo " "
echo "Table of $n"
echo " "
i=1
while [ $i -le 10 ]
do
res=`expr $i \* $n`
echo "$n X $i = $res"
i=`expr $i + 1`

done
done
```



```
cdac@LAPTOP-HJG23I52:~$ nano AssignmentQues.sh
cdac@LAPTOP-HJG23I52:~$ bash AssignmentQues.sh

Table of 1:

1 X 1 = 1
1 X 2 = 2
1 X 3 = 3
1 X 4 = 4
1 X 5 = 5
1 X 6 = 6
1 X 7 = 7
1 X 8 = 8
1 X 9 = 9
1 X 10 = 10

Table of 2:

2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 10
2 X 6 = 12
2 X 7 = 14
2 X 8 = 16
2 X 9 = 18
2 X 10 = 20
```

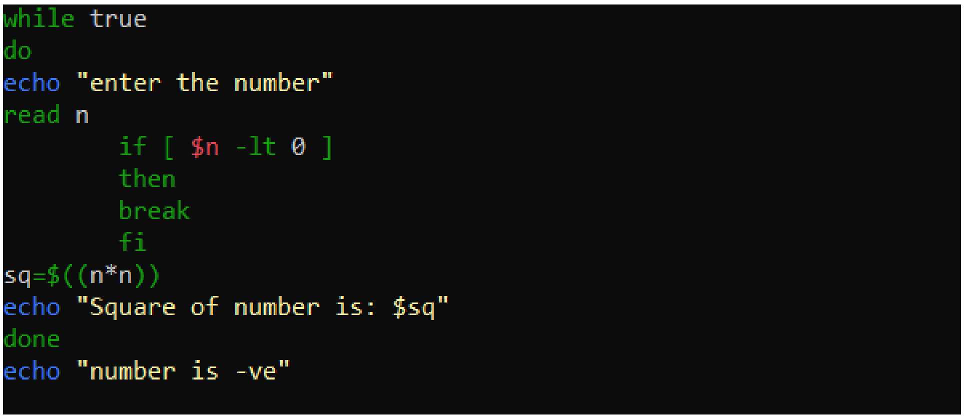


```
Table of 3:

3 X 1 = 3
3 X 2 = 6
3 X 3 = 9
3 X 4 = 12
3 X 5 = 15
3 X 6 = 18
3 X 7 = 21
3 X 8 = 24
3 X 9 = 27
3 X 10 = 30

Table of 4:

4 X 1 = 4
4 X 2 = 8
4 X 3 = 12
4 X 4 = 16
4 X 5 = 20
4 X 6 = 24
4 X 7 = 28
4 X 8 = 32
4 X 9 = 36
4 X 10 = 40
```

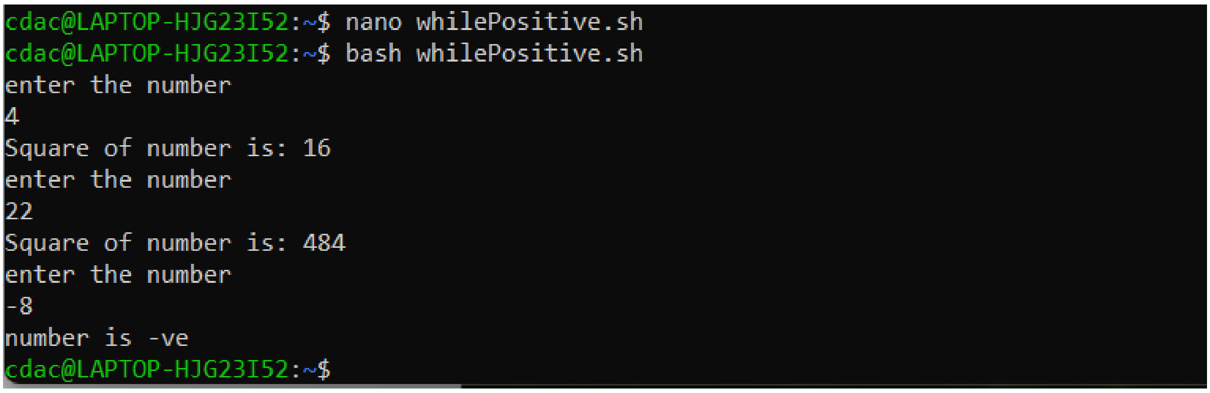


```
Table of 5:

5 X 1 = 5
5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
5 X 10 = 50
```

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.



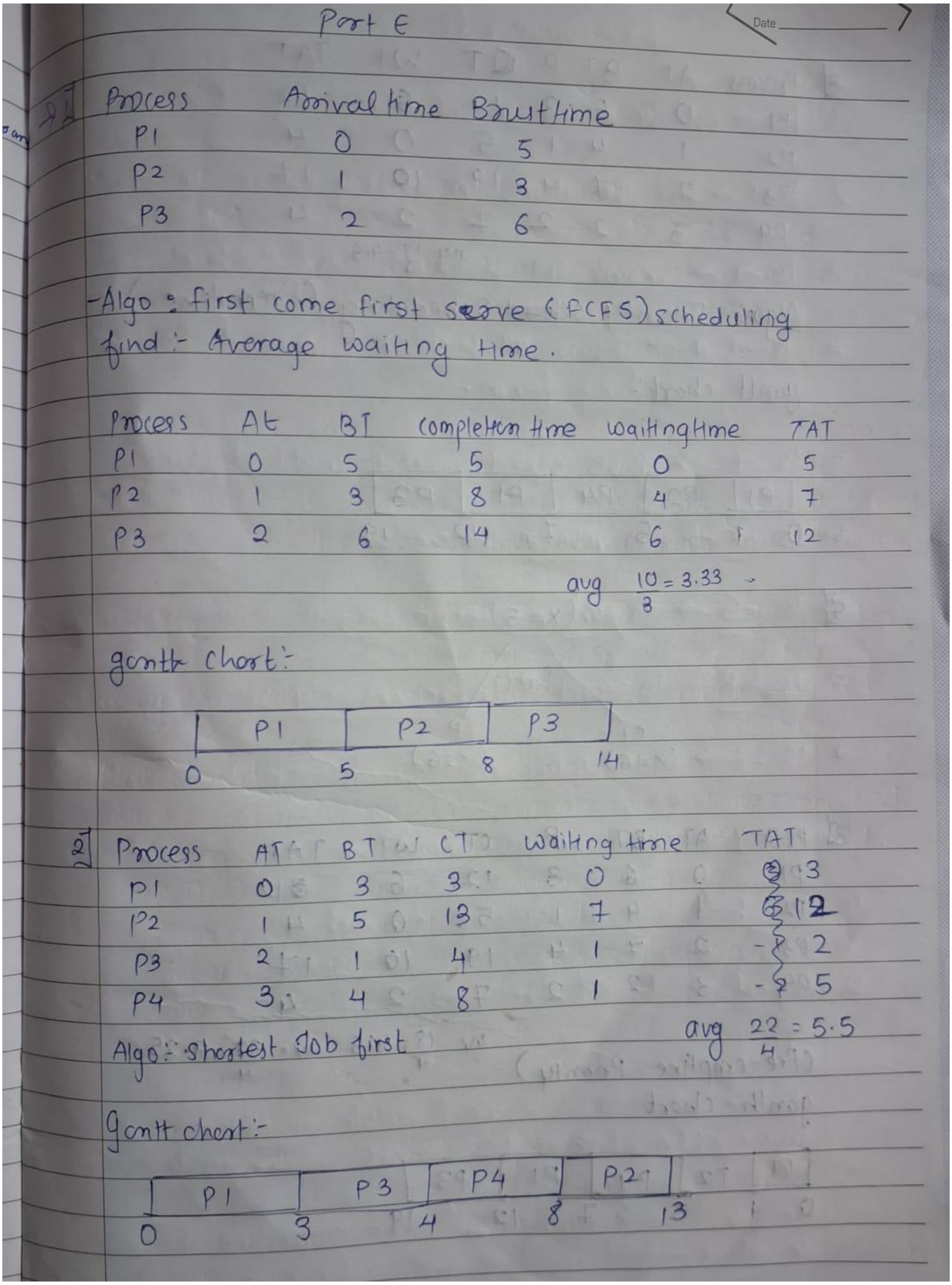
```
while true
do
echo "enter the number"
read n
        if [ $n -lt 0 ]
        then
        break
        fi
sq=$((n*n))
echo "Square of number is: $sq"
done
echo "number is -ve"
```



```
cdac@LAPTOP-HJG23I52:~$ nano whilePositive.sh
cdac@LAPTOP-HJG23I52:~$ bash whilePositive.sh
enter the number
4
Square of number is: 16
enter the number
22
Square of number is: 484
enter the number
-8
number is -ve
cdac@LAPTOP-HJG23I52:~$
```

Part E

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



Part E

Date

1]

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

-Algo : first come first serve (FCFS) scheduling
find :- Average waiting time.

Process	At	BT	completion time	waiting time	TAT
P1	0	5	5	0	5
P2	1	3	8	4	7
P3	2	6	14	6	12

avg $\frac{10}{3} = 3.33$

gantt chart :-

P1	P2	P3

0 5 8 14

2]

Process	AT	BT	CT	waiting time	TAT
P1	0	3	3	0	3
P2	1	5	13	7	12
P3	2	1	4	1	2
P4	3	4	8	1	5

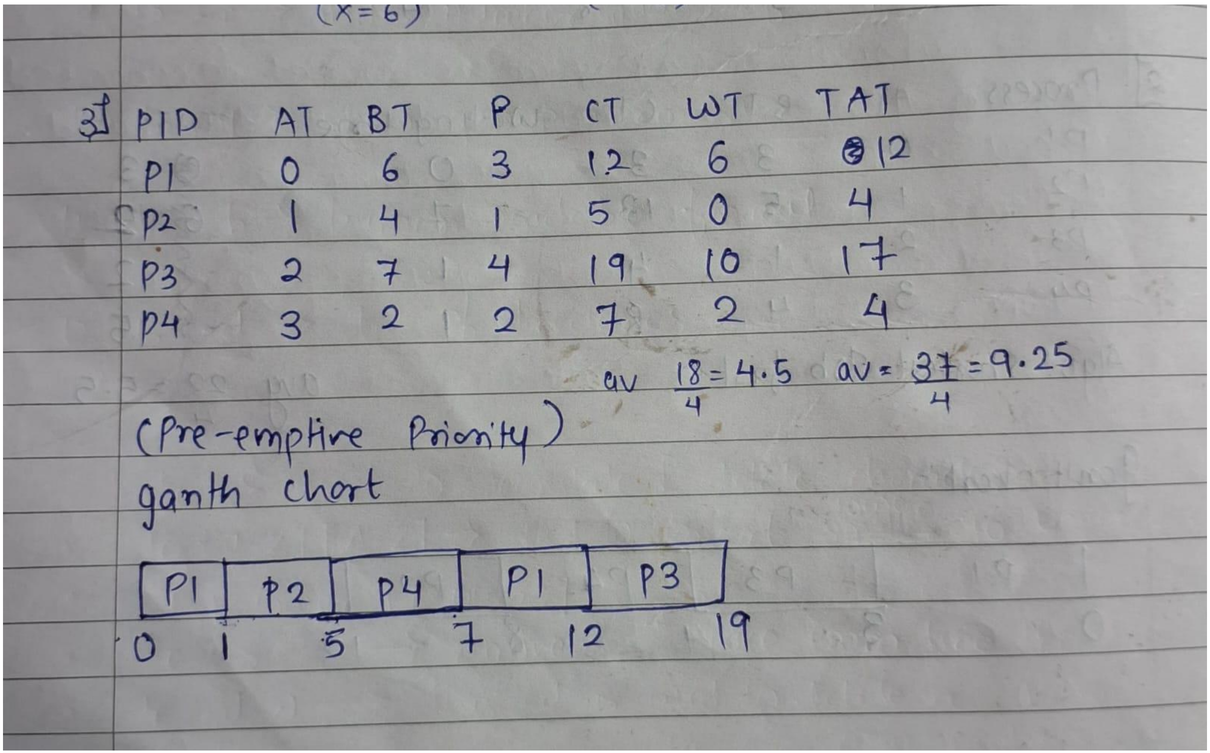
avg $\frac{22}{4} = 5.5$

Algo :- shortest Job first

gantt chart :-

P1	P3	P4	P2

0 3 4 8 13



(x=6)

3]

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

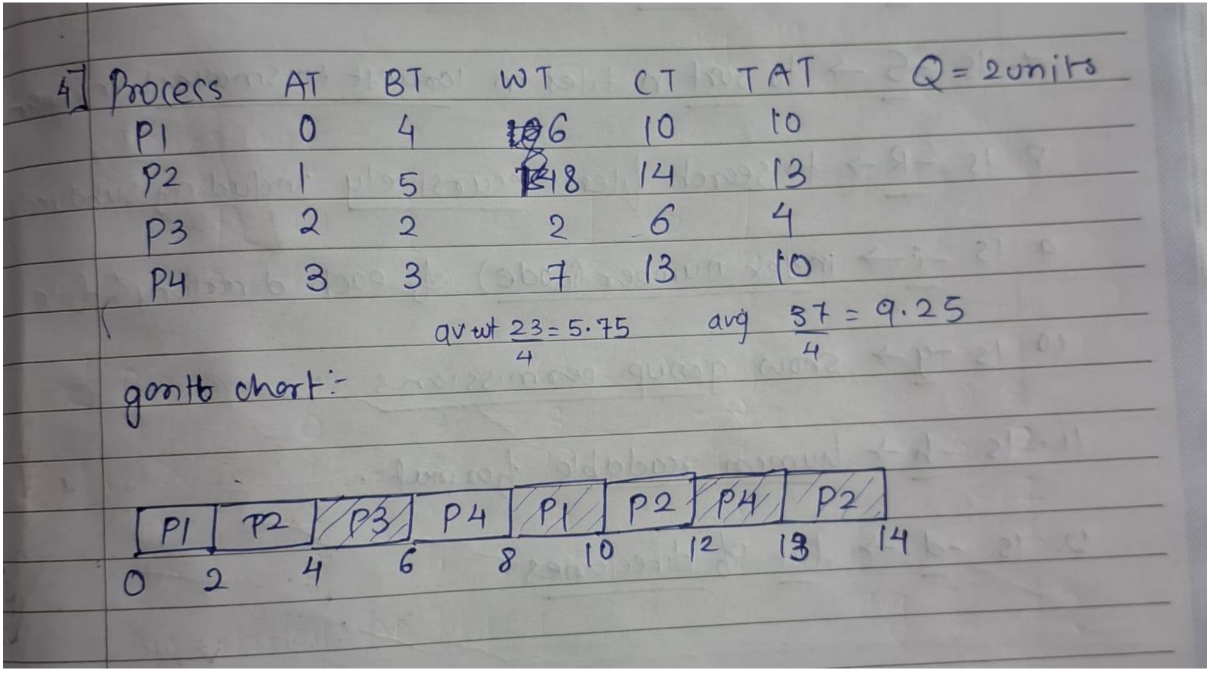
av $\frac{18}{4} = 4.5$ av = $\frac{37}{4} = 9.25$

(Pre-emptive Priority)

ganth chart

P1	P2	P4	P1	P3

0 1 5 7 12 19



4] Q = 2units

Process	AT	BT	WT	CT	TAT
P1	0	4	~~10~~6	10	10
P2	1	5	~~14~~8	14	13
P3	2	2	2	6	4
P4	3	3	7	13	10

av wt $\frac{23}{4} = 5.75$ avg $\frac{37}{4} = 9.25$

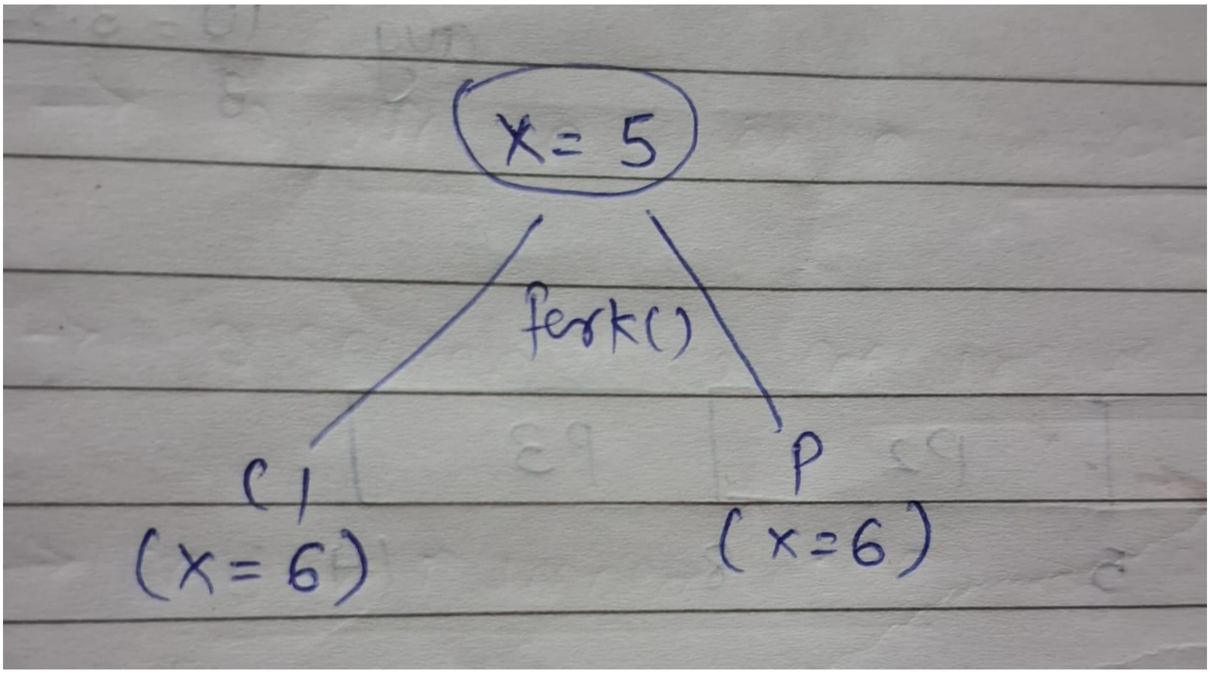
gantt chart:-

P1	P2	P3	P4	P1	P2	P4	P2

0 2 4 6 8 10 12 13 14

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?

Sol: As fork is used to create a child process of the parent process and the child process does the same thing as that of the parent process and hence the parent and child will both have x=6.


X= 5
fork()
C1
(X = 6)
P
(x=6)