COS Assignment 2

Mansi Mahadev Kamble

Part A

Q.What will the following commands do?

1.echo "Hello, World!"
Prints "Hello, World!" to the terminal.

2.name="Productive"
Assigns "Productive" to the variable name.

3.touch file.txt
Creates an empty file named file.txt or updates its timestamp if it exists .

4.ls -a
Lists all files and directories, including hidden ones those starting with .

5.rm file.txt
Deletes file.txt permanently.

6.cp file1.txt file2.txt
Copies file1.txt to file2.txt

7.mv file.txt /path/to/directory/
Moves file.txt to the specified directory.

8.chmod 755 script.sh
Grants the owner full permissions (read, write, execute) and gives others read and execute permissions on script.sh.

9.grep "pattern" file.txt
Searches for "pattern" in file.txt and displays matching lines.

10.kill PID
Terminates the process with the given process ID (PID).

11.mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
Creates mydir, moves into it, creates file.txt, writes "Hello, World!" to it, and displays its contents.

12.ls -l | grep ".txt"
Lists files in long format and filters entries containing .txt.

13.cat file1.txt file2.txt | sort | uniq
Merges file1.txt and file2.txt, sorts the lines, and removes duplicates

14.ls -l | grep "^d"
Lists only directories in long format ,^d filters lines starting with d, indicating directories.

15.grep -r "pattern" /path/to/directory/
Recursively searches for "pattern" in all files under the specified directory.

16.cat file1.txt file2.txt | sort | uniq -d
Displays only duplicate lines found in file1.txt and file2.txt after sorting.

17.chmod 644 file.txt
Grants the owner read and write permissions, while others get read-only access to file.txt.

18.cp -r source_directory destination_directory
Recursively copies source_directory and its contents to destination_directory.

19.chmod u+x file.txt
Grants execute permission to the file owner.

20.echo $PAT
Displays the value of the environment variable PAT (if set). Otherwise, it prints an empty line.

21.find /path/to/search -name "*.txt"
Searches for all .txt files within the specified path.

Part B

Q.Identify True or False

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

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

2.mv is used to move files and directories.

True – mv is used to move files and directories.

3.cd is used to copy files and directories.

False – cd is used to change the directory, not copy files. (cp is used for copying files and directories.)

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

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

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

True – 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.

True – chmod 755 file.txt gives: Owner: read (r), write (w), execute (x)

Group: read (r), execute (x)

Others: read (r), execute (x)

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

True – mkdir -p directory1/directory2 creates directory1 first (if it doesn't exist), then directory2 inside it.

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

True – rm -rf file.txt forcefully removes the file without confirmation (-r is for recursive removal, and -f is for force).

Q.Identify the Incorrect Commands:

1.chmodx is used to change file permissions.

Incorrect – chmodx does not exist. The correct command is chmod.

2.cpy is used to copy files and directories.

Incorrect – cpy does not exist. The correct command is cp.

3.mkfile is used to create a new file.

Incorrect – mkfile is not a standard Linux command. To create a file, use touch filename or echo "" > filename.

4.catx is used to concatenate files.

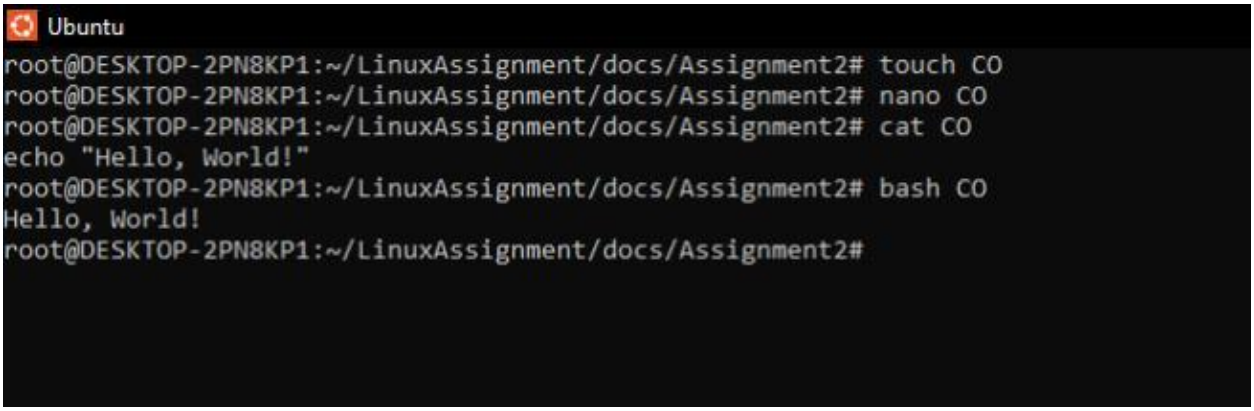
Incorrect – catx does not exist. The correct command for concatenation is cat.

5.rn is used to rename Files.

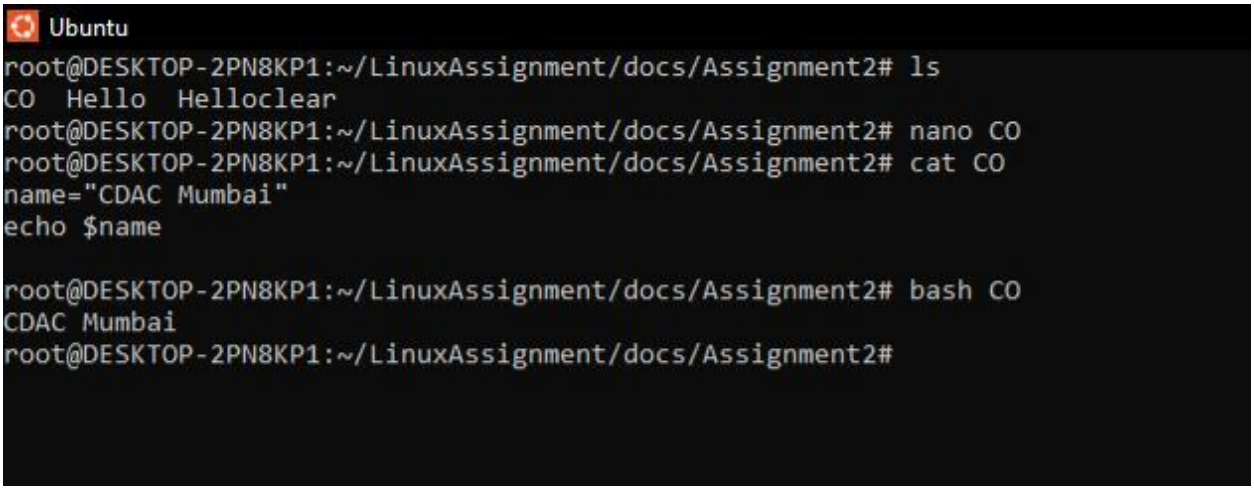
Incorrect – rn does not exist. The correct command to rename files is mv.

Part C

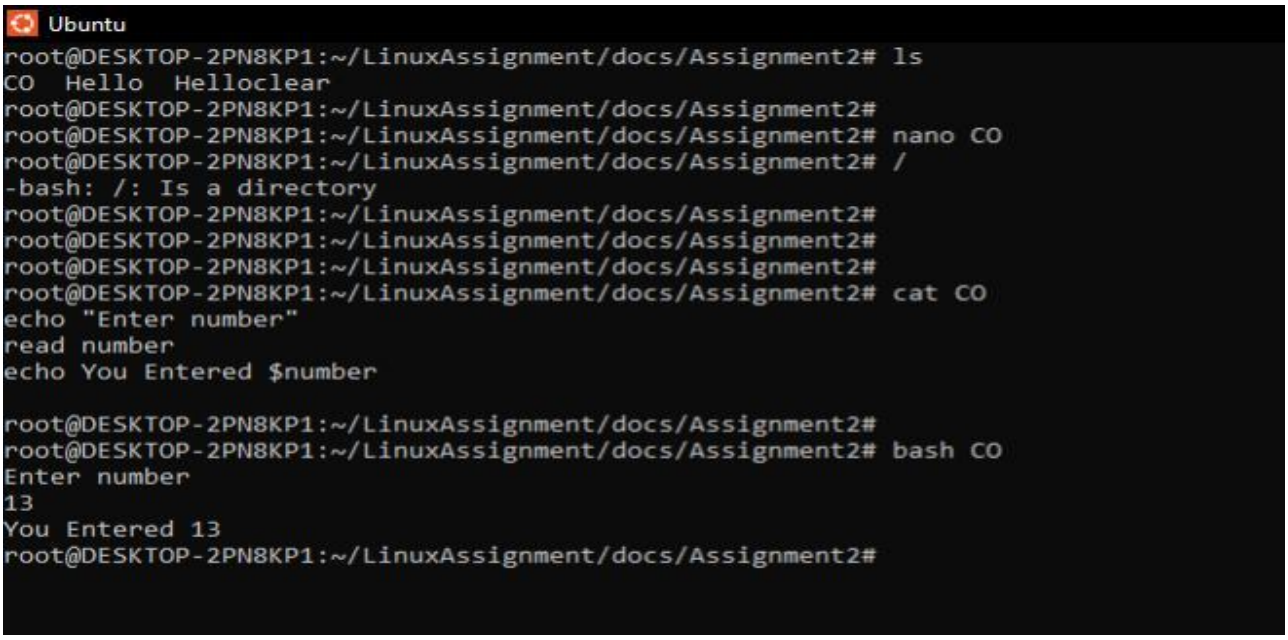
Question 1: Write a shell script that prints "Hello, World!" to the terminal.



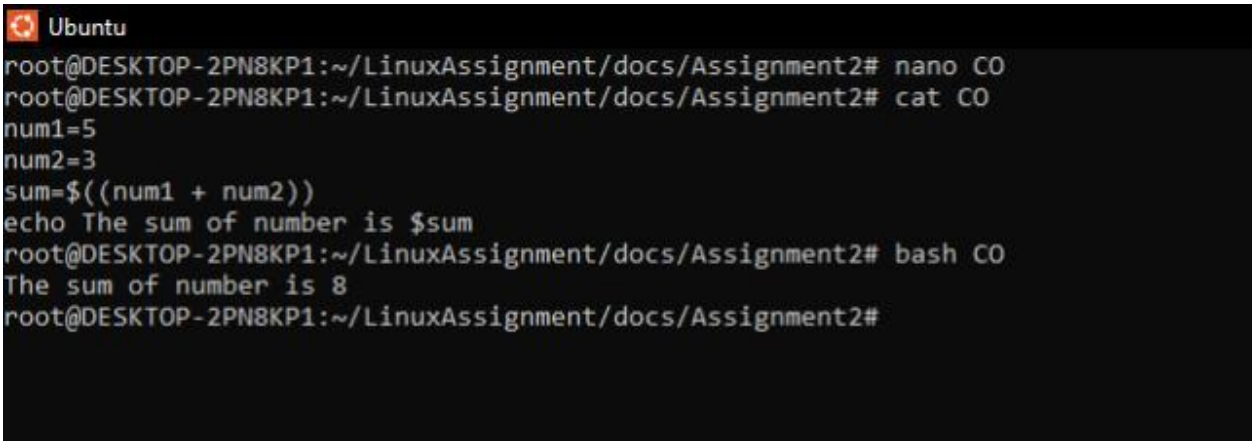
Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" Print the value of variable



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

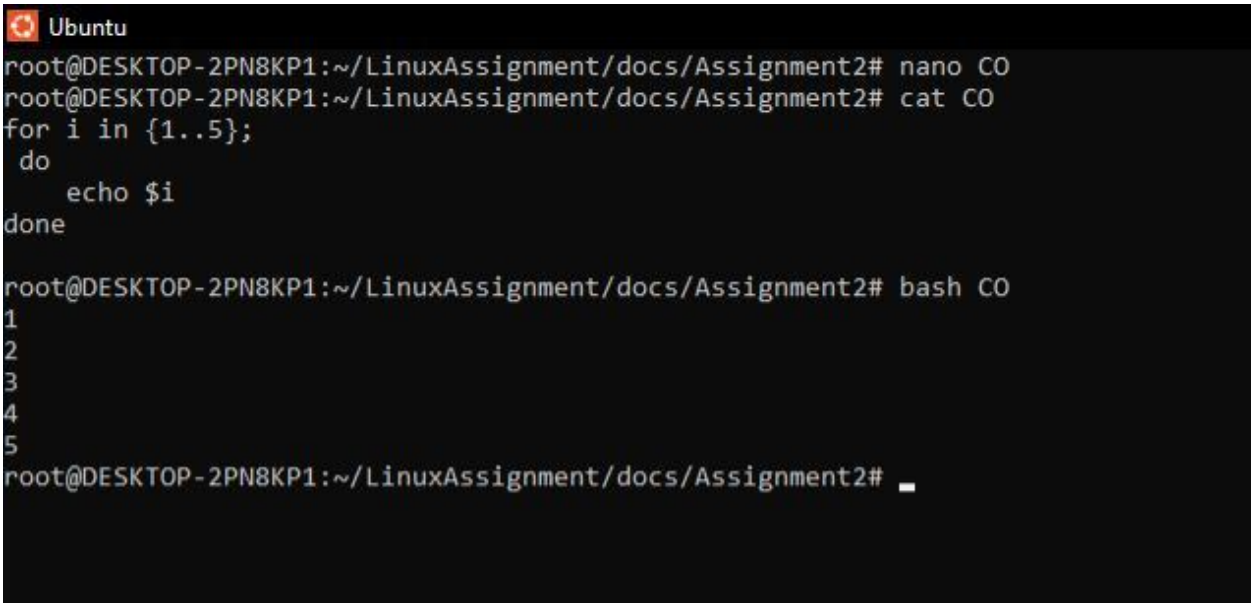


Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.

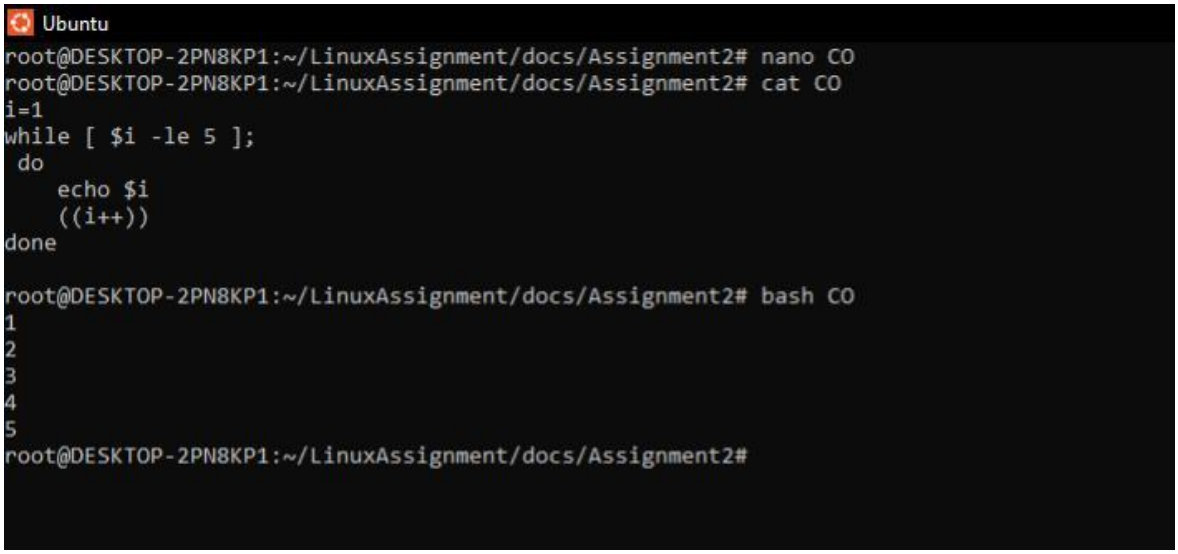


Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

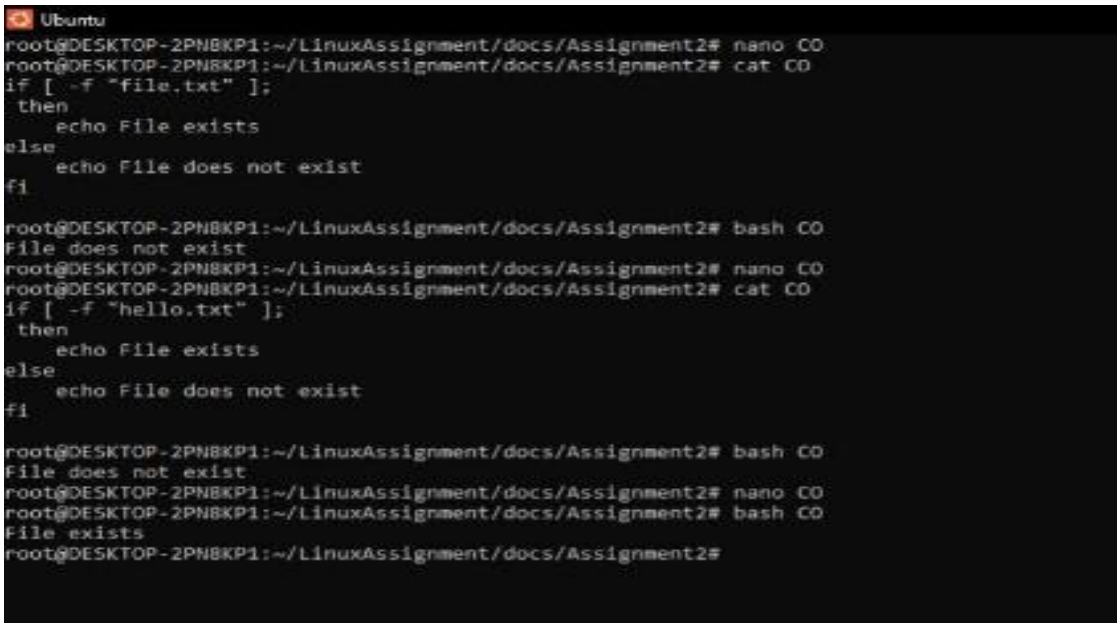
Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.



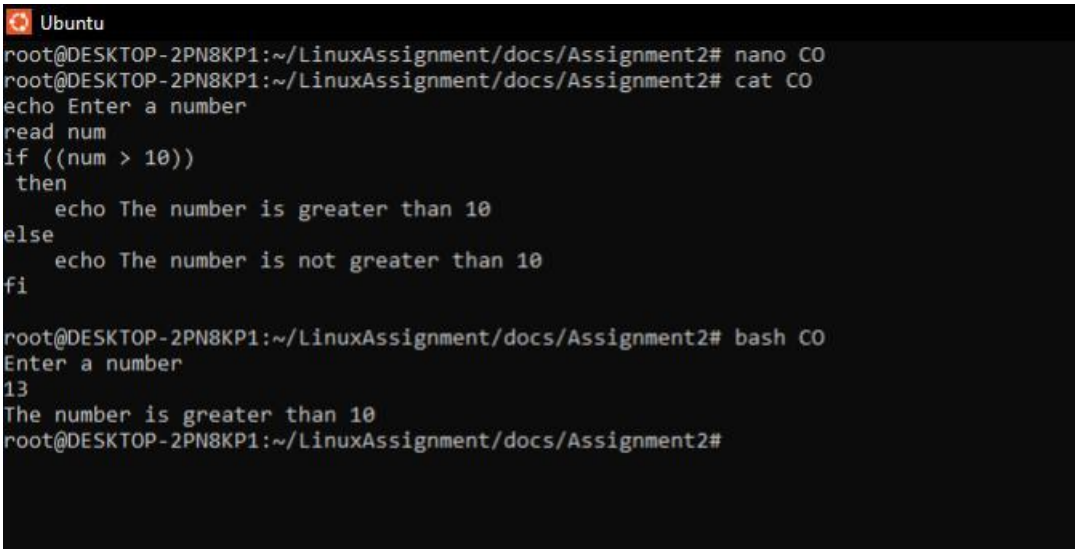
Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.



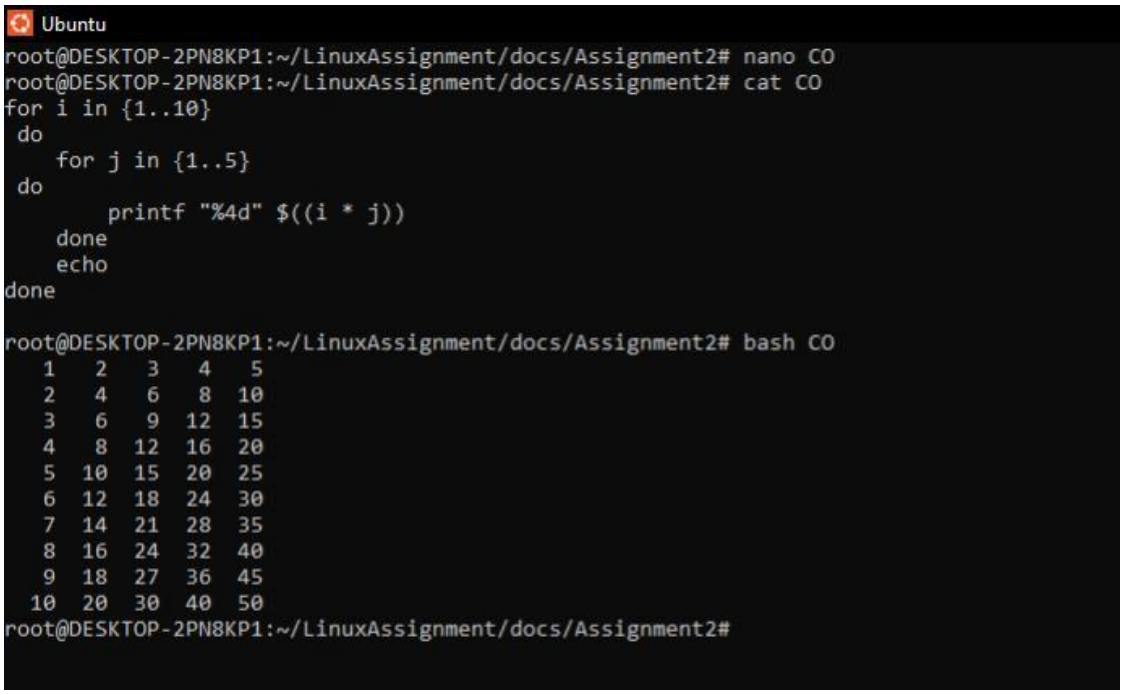
Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".



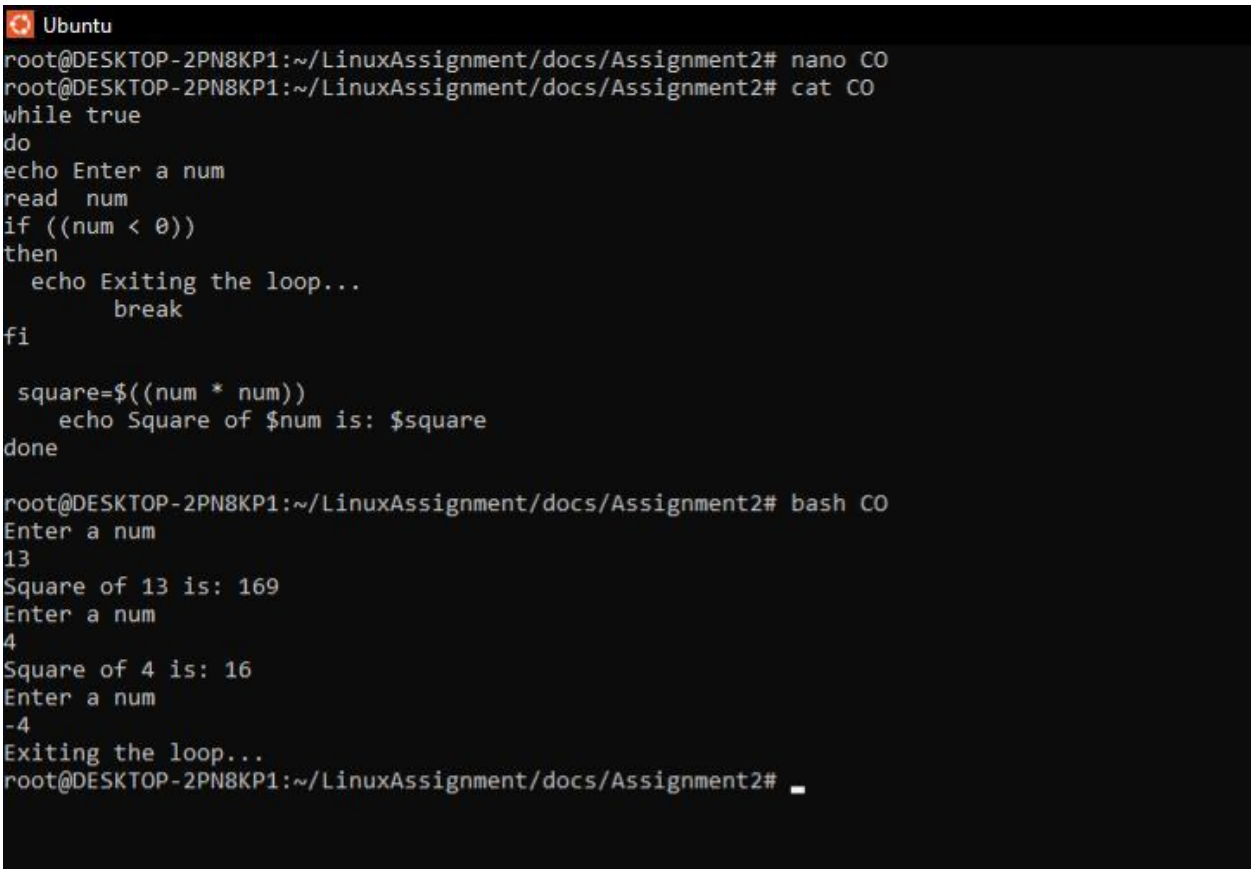
Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.



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



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.



Part E

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

| Process | Arrival Time | Burst Time |

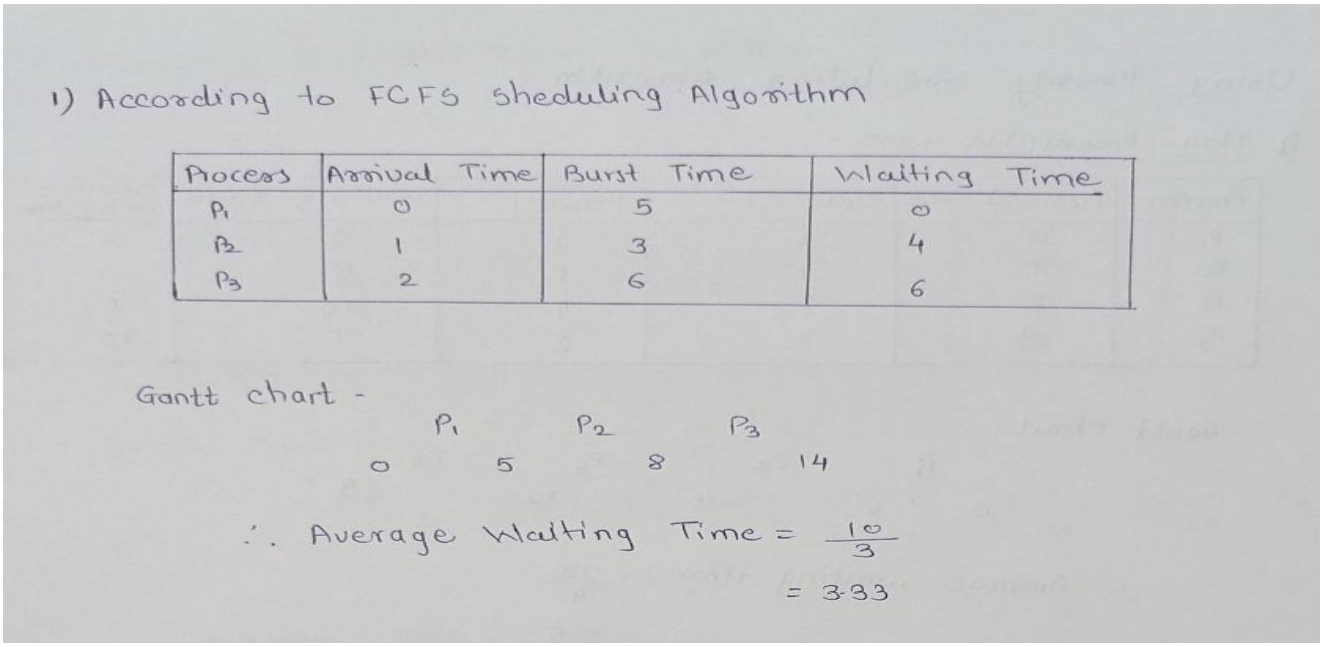
|---------|--------------|------------|

| P1 | 0 | 5 |

| P2 | 1 | 3 |

| P3 | 2 | 6 |

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



1) According to FCFS Scheduling Algorithm

Process	Arrival Time	Burst Time	Waiting Time
P_1	0	5	0
P_2	1	3	4
P_3	2	6	6

Gantt chart -

P_1 | P_2 | P_3

0 5 8 14

∴ Average Waiting Time = $\frac{10}{3}$

= 3.33

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

| Process | Arrival Time | Burst Time |

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

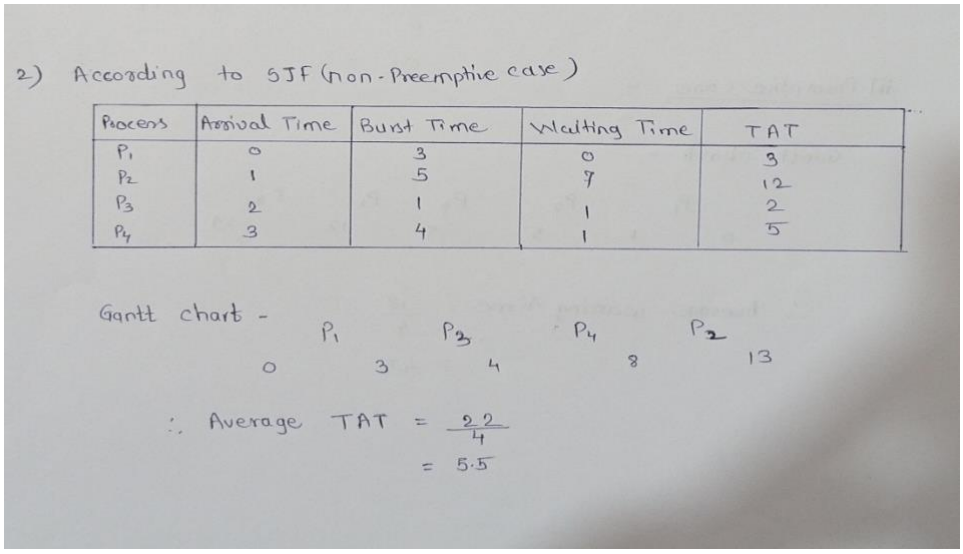
| P1 | 0 | 3 |

| P2 | 1 | 5 |

| P3 | 2 | 1 |

| P4 | 3 | 4 |

Calculate the average turnaround time



2) According to SJF (non-Preemptive case)

Process	Arrival Time	Burst Time	Waiting Time	TAT
P_1	0	3	0	3
P_2	1	5	7	12
P_3	2	1	1	2
P_4	3	4	1	5

Gantt chart -

P_1 | P_3 | P_4 | P_2

0 3 4 8 13

∴ Average TAT = $\frac{22}{4}$

= 5.5

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

indicates higher priority):

| Process | Arrival Time | Burst Time | Priority |

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

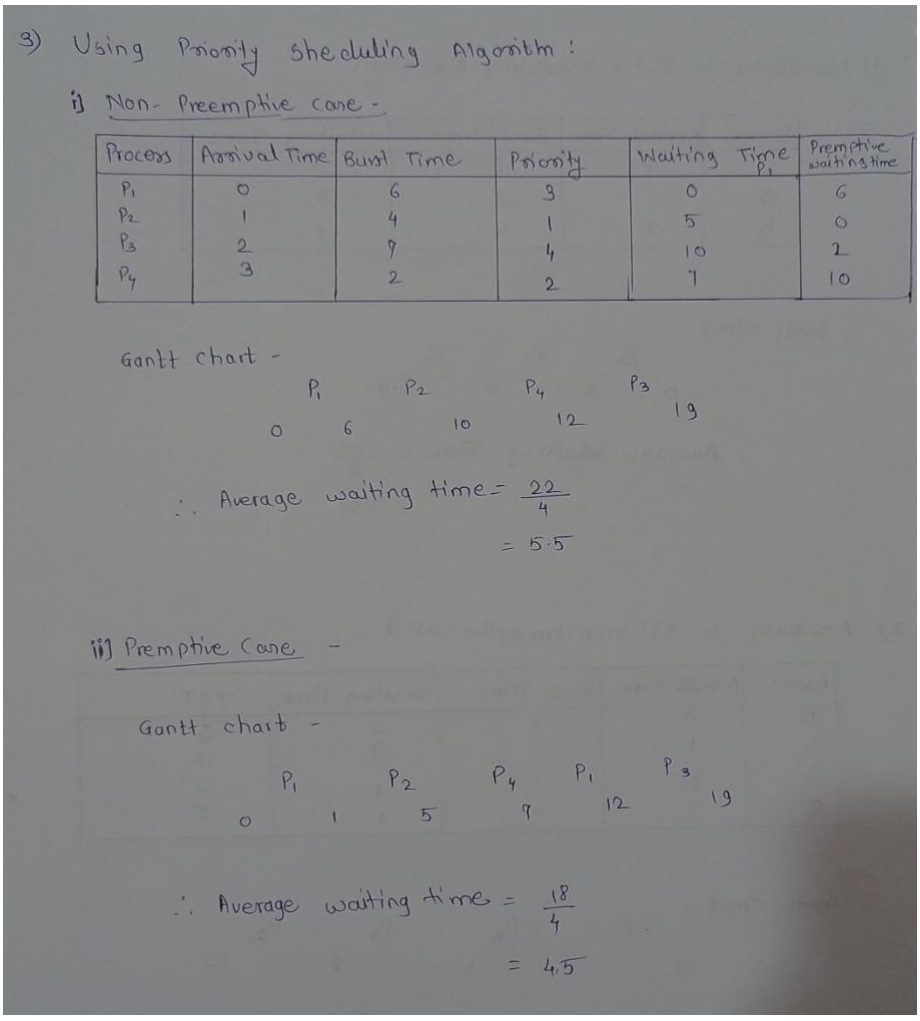
| P1 | 0 | 6 | 3 |

| P2 | 1 | 4 | 1 |

| P3 | 2 | 7 | 4 |

| P4 | 3 | 2 | 2 |

Calculate the average waiting time using Priority Scheduling.



3) Using Priority Scheduling Algorithm :

i) Non- Preemptive case -

Process	Arrival Time	Burst Time	Priority	Waiting Time P_i	Premptive waiting time
P_1	0	6	3	0	6
P_2	1	4	1	5	0
P_3	2	7	4	10	2
P_4	3	2	2	7	10

Gantt chart -

P_1 | P_2 | P_4 | P_3

0 6 10 12 19

∴ Average waiting time = $\frac{22}{4}$

$= 5.5$

ii) Premptive Case -

Gantt chart -

P_1 | P_2 | P_4 | P_1 | P_3

0 1 5 7 12 19

∴ Average waiting time = $\frac{18}{4}$

$= 4.5$

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 |

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

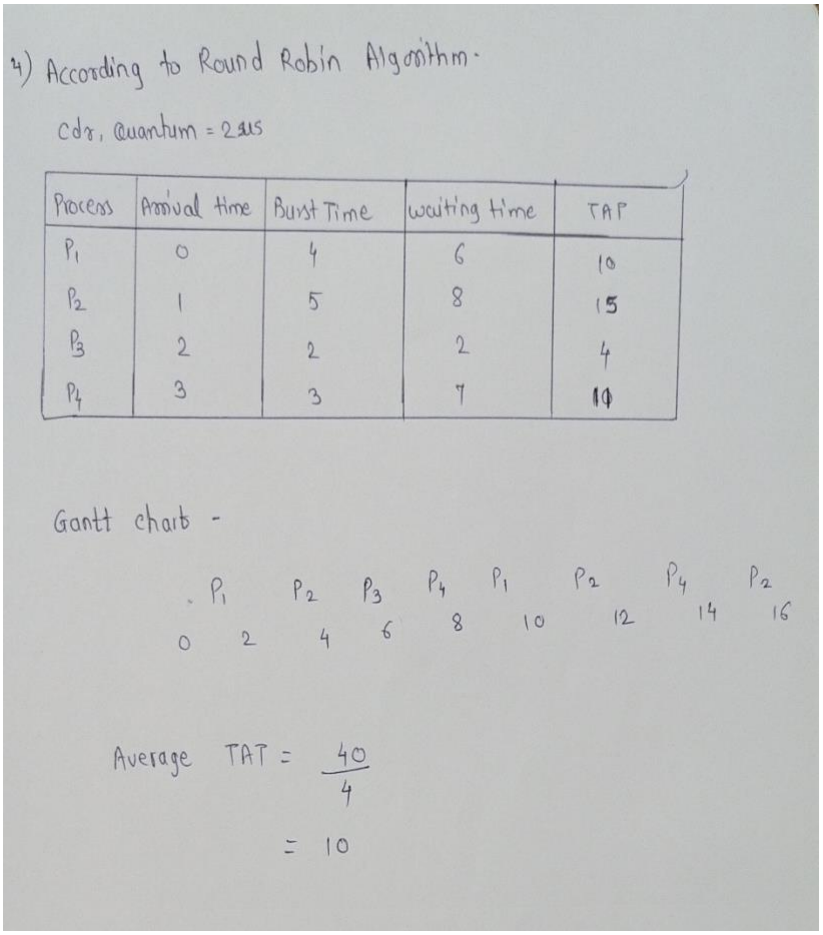
| P1 | 0 | 4 |

| P2 | 1 | 5 |

| P3 | 2 | 2 |

| P4 | 3 | 3 |

Calculate the average turnaround time using Round Robin scheduling.



4) According to Round Robin Algorithm.

Cdr, Quantum = 2 sec

Process	Arrival time	Burst Time	waiting time	TAT
P_1	0	4	6	10
P_2	1	5	8	15
P_3	2	2	2	4
P_4	3	3	7	10

Gantt chart -

P_1	P_2	P_3	P_4	P_1	P_2	P_4	P_2
0	2	4	6	8	10	12	14	16

Average TAT = $\frac{40}{4}$

= 10

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?round time using Shortest Job First (SJF) scheduling.

Answer :

When a process calls fork(), it creates a child process that is a duplicate of the parent. Both processes have separate memory spaces, meaning any changes to variables in one process do not affect the other.

In this case:

- Initially, x = 5 in the parent process.
- After fork(), both parent and child have separate copies of x=5.
- Both processes increment x by 1 independently.

Final values:

Parent process: x = 6

Child process: x = 6