Assignment 2

Part A

Prajyot Nimsarkar

echo "Hello, World!" **:** Prints "Hello, World!" on the cmd terminal.

name="Productive" : Assigns the string "Productive" to the variable name. it will be saved until the system is not restarted

touch file.txt : Creates an empty file named file.txt. in the current directory

ls -a :-Lists all files, including hidden ones / list information about the files in the current directory.

rm file.txt : Deletes the file file.txt in that directory .

cp file1.txt file2.txt : Copy the content of file1.txt to the file2.txt.

mv file.txt /path/to/directory/ : Move the file.txt file to the specified directory that we will given in cmd.

chmod 755 script.sh :- Gives read, write, execute permissions to the owner and read,execute permission to group and others for that file.

grep "pattern" file.txt :- Searches for "pattern" in file.txt.

kill PID : Terminates the process with the given PID orProcess ID .

mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt :- Creates a directory mydir, enters in mydir directory then , creates file.txt then , writes "Hello, World!" into the file.txt, and then displays the contents of file.txt.

ls -l | grep ".txt" :- Lists all .txt files in long format.

cat file1.txt file2.txt | sort | uniq – Merges the both files then sorts, and removes duplicate lines from file1.txt and file2.txt and show the uniqe values.

ls -l | grep "^d" – Lists only directories in that directory .

grep -r "pattern" /path/to/directory/ :- Searches for "pattern" recursively in the given directory.

cat file1.txt file2.txt | sort | uniq -d :- Shows duplicate lines common to both files in the file1.txt & file2.txt.

chmod 644 file.txt :- Gives the owner read &write permission and gives read-only permissions to the group and others.

cp -r source_directory destination_directory :- Copies a directory recursively.

find /path/to/search -name "*.txt" :- Searches for .txt files in the given path.

chmod u+x file.txt :- Gives execute permission to the file owner of file.txt .

echo $PATH :- Displays the system's executable search paths in the cmd.

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.

chmod

2. cpy is used to copy files and directories.

cp

3. mkfile is used to create a new file.

Touch filename

4. catx is used to concatenate files.

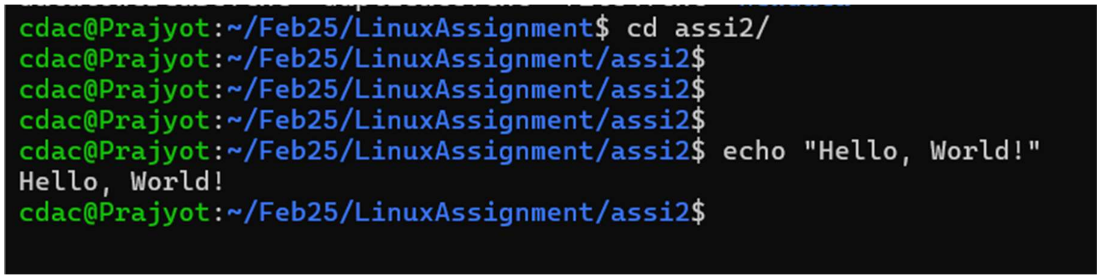
cat

5. rn is used to rename files.

mv oldname newname

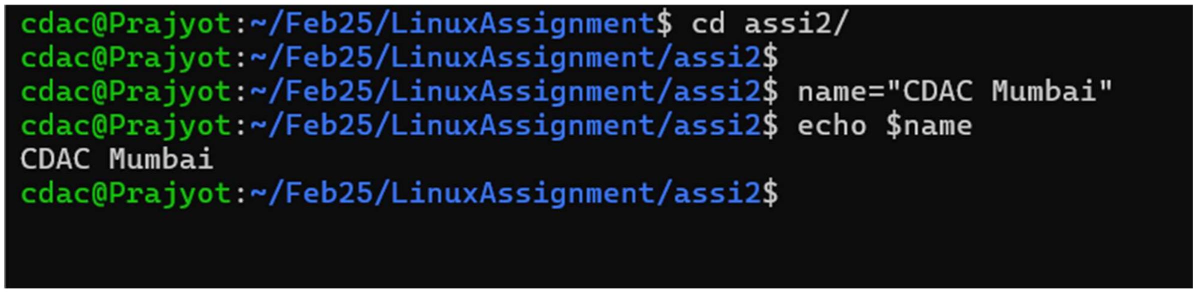
Part C

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



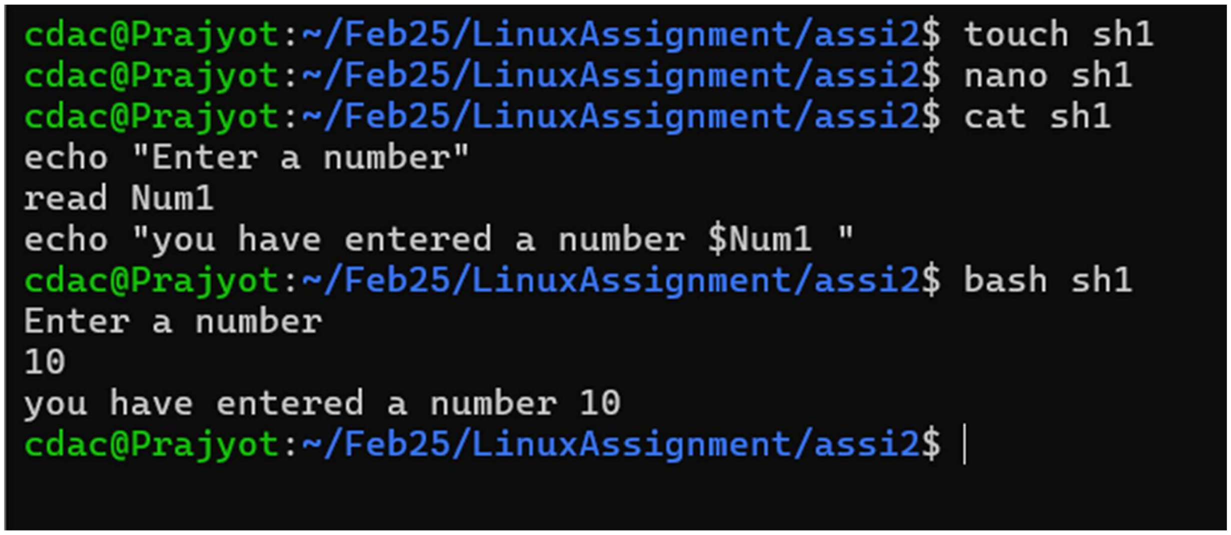
```
cdac@Prajyot:~/Feb25/LinuxAssignment$ cd assi2/
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ echo "Hello, World!"
Hello, World!
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$
```

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



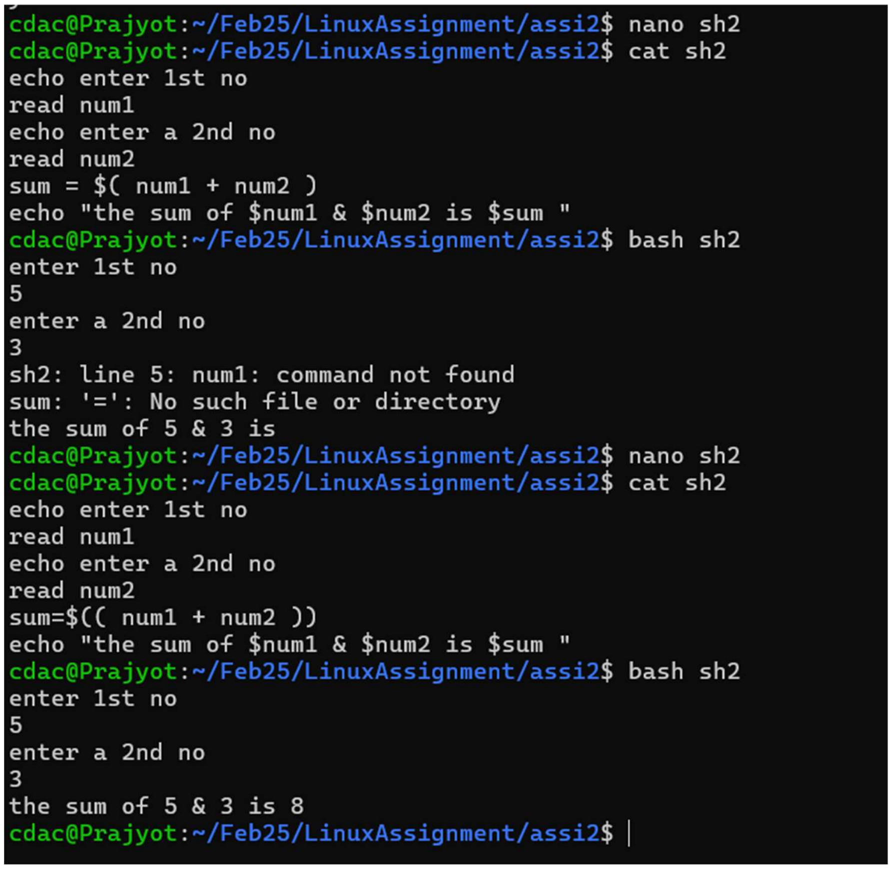
```
cdac@Prajyot:~/Feb25/LinuxAssignment$ cd assi2/
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ name="CDAC Mumbai"
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ echo $name
CDAC Mumbai
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$
```

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



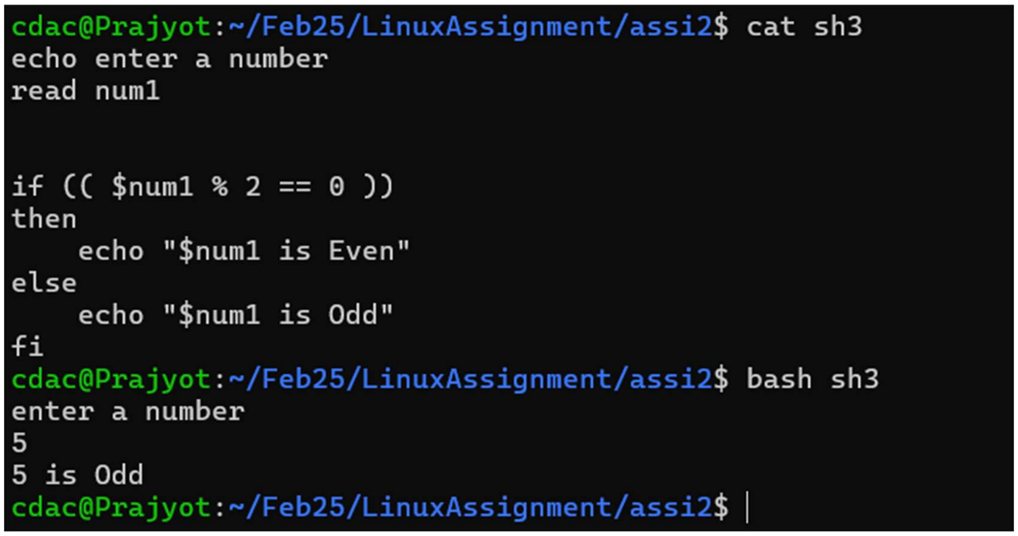
```
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ touch sh1
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ nano sh1
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ cat sh1
echo "Enter a number"
read Num1
echo "you have entered a number $Num1 "
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ bash sh1
Enter a number
10
you have entered a number 10
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$
```

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



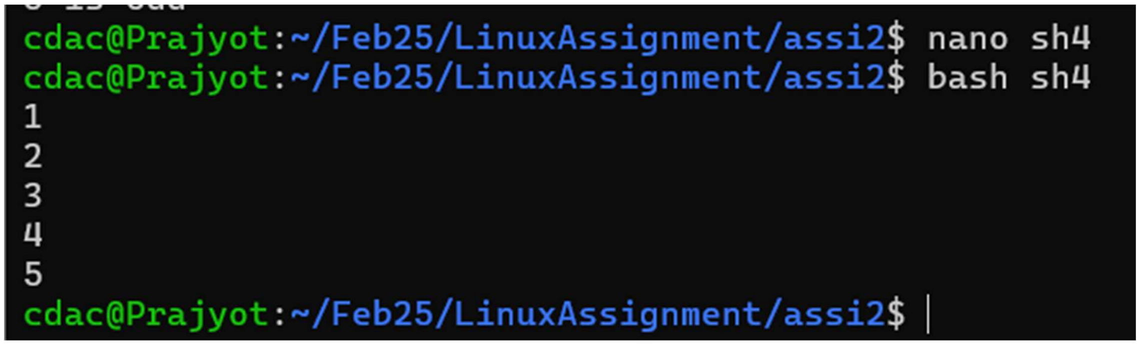
```
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ nano sh2
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ cat sh2
echo enter 1st no
read num1
echo enter a 2nd no
read num2
sum = $( num1 + num2 )
echo "the sum of $num1 & $num2 is $sum "
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ bash sh2
enter 1st no
5
enter a 2nd no
3
sh2: line 5: num1: command not found
sum: '=': No such file or directory
the sum of 5 & 3 is
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ nano sh2
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ cat sh2
echo enter 1st no
read num1
echo enter a 2nd no
read num2
sum=$(( num1 + num2 ))
echo "the sum of $num1 & $num2 is $sum "
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ bash sh2
enter 1st no
5
enter a 2nd no
3
the sum of 5 & 3 is 8
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$
```

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



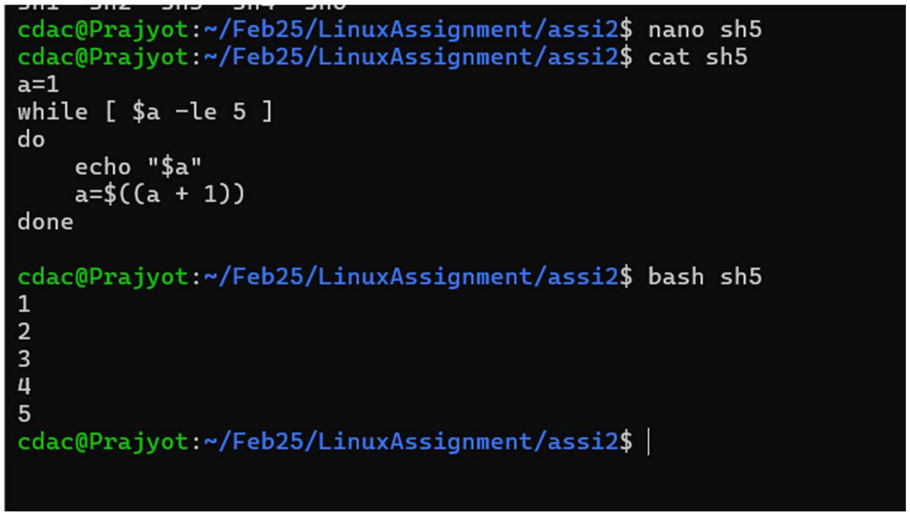
```
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ cat sh3
echo enter a number
read num1


if (( $num1 % 2 == 0 ))
then
    echo "$num1 is Even"
else
    echo "$num1 is Odd"
fi
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ bash sh3
enter a number
5
5 is Odd
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$
```

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



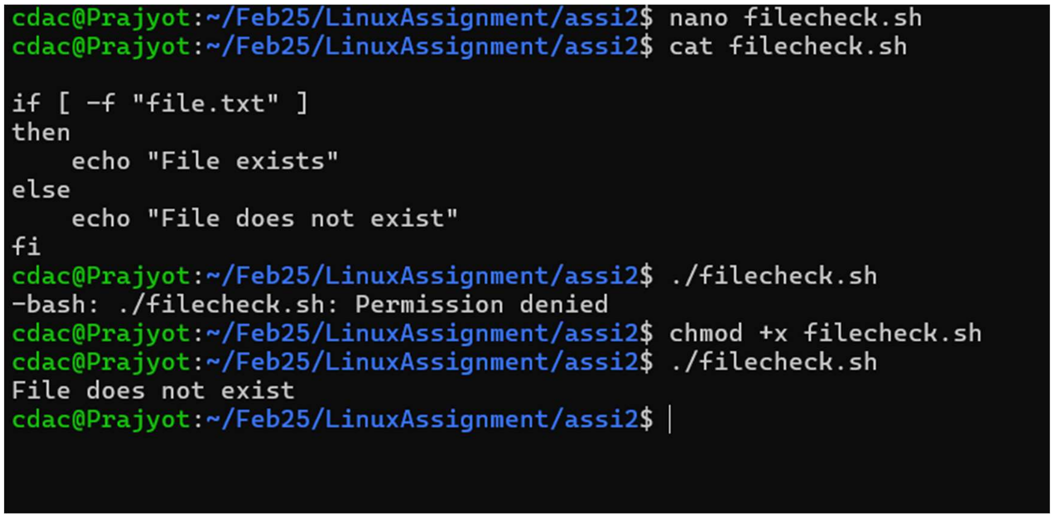
```
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ nano sh4
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ bash sh4
1
2
3
4
5
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ |
```

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



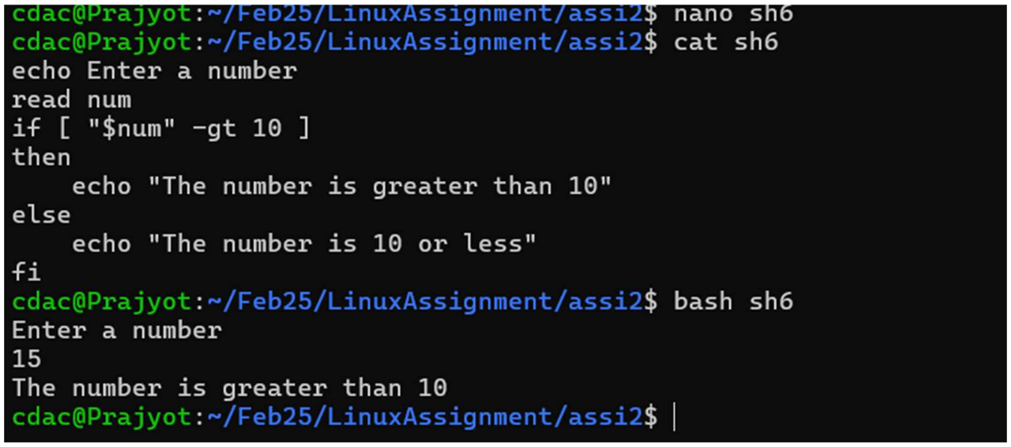
```
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ nano sh5
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ cat sh5
a=1
while [ $a -le 5 ]
do
    echo "$a"
    a=$((a + 1))
done

cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ bash sh5
1
2
3
4
5
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ |
```

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".



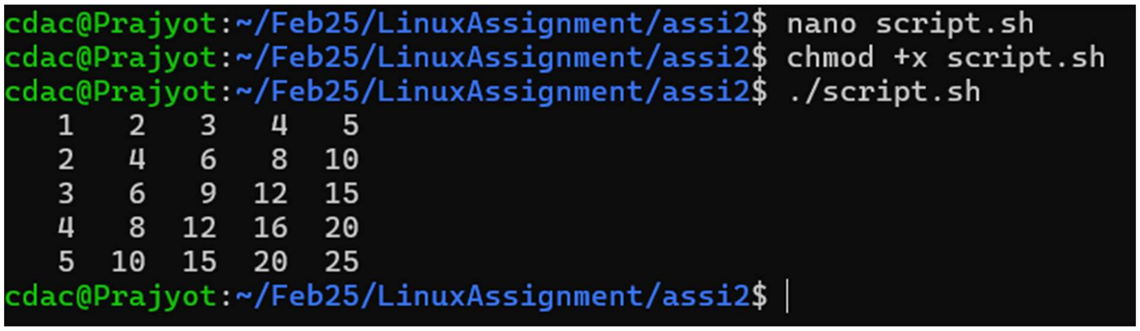
```
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ nano filecheck.sh
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ cat filecheck.sh

if [ -f "file.txt" ]
then
    echo "File exists"
else
    echo "File does not exist"
fi
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ ./filecheck.sh
-bash: ./filecheck.sh: Permission denied
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ chmod +x filecheck.sh
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ ./filecheck.sh
File does not exist
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ |
```

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.



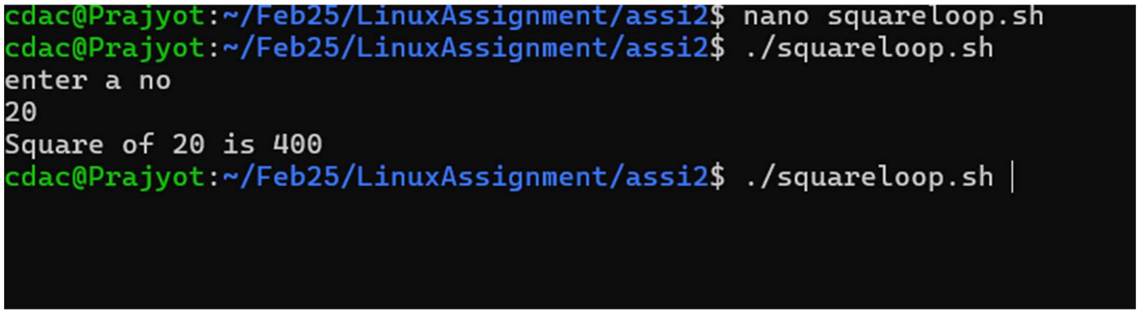
```
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ nano sh6
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ cat sh6
echo Enter a number
read num
if [ "$num" -gt 10 ]
then
    echo "The number is greater than 10"
else
    echo "The number is 10 or less"
fi
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ bash sh6
Enter a number
15
The number is greater than 10
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ 
```

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@Prajyot:~/Feb25/LinuxAssignment/assi2$ nano script.sh
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ chmod +x script.sh
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ ./script.sh
   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@Prajyot:~/Feb25/LinuxAssignment/assi2$ 
```

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



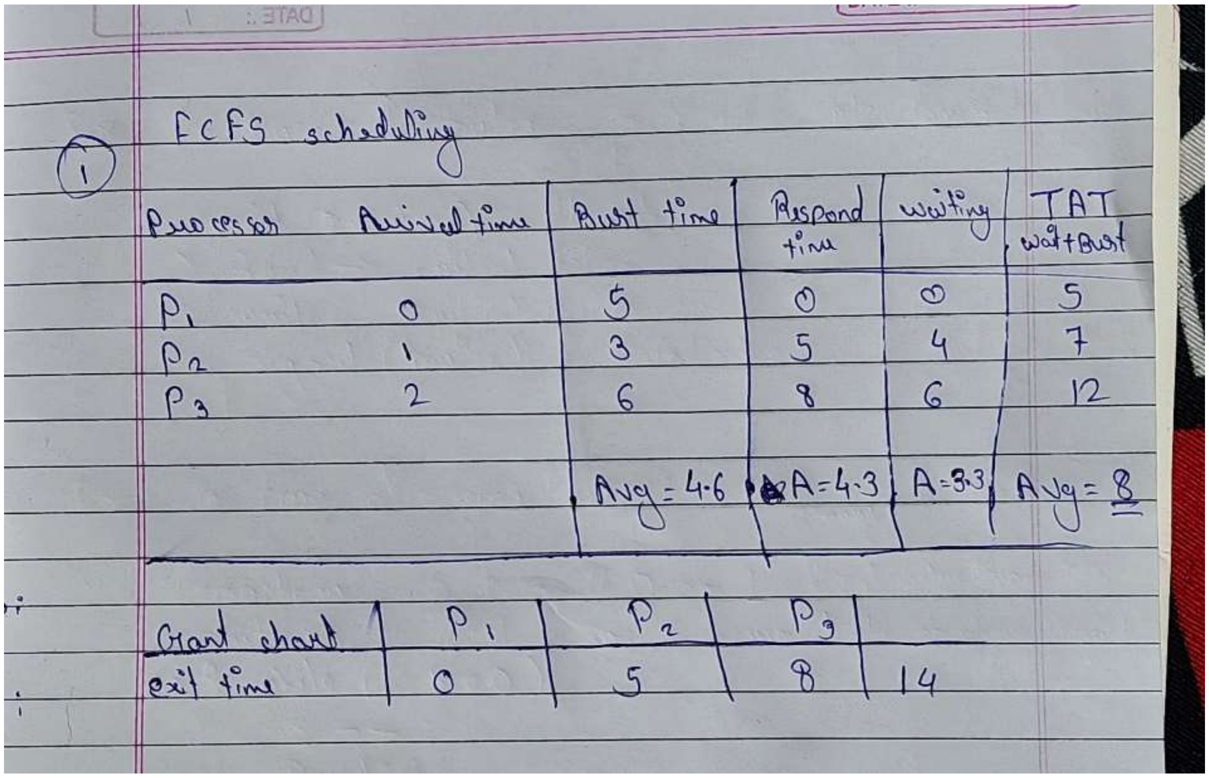
```
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ nano squareloop.sh
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ ./squareloop.sh
enter a no
20
Square of 20 is 400
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ ./squareloop.sh 
```

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) FCFS scheduling

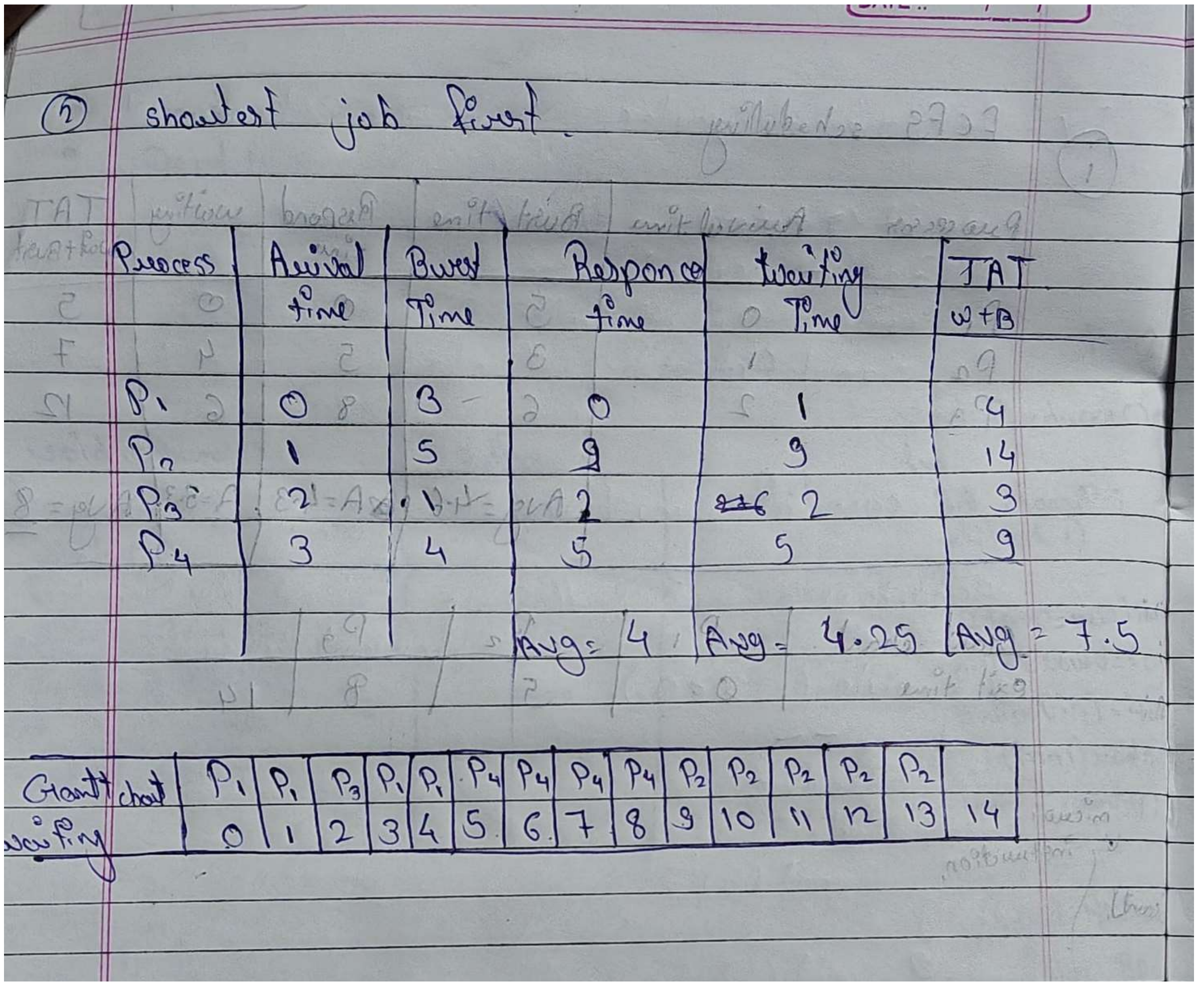
Processor	Arrival time	Burst time	Respond time	waiting	TAT wait+Burst
P_1	0	5	0	0	5
P_2	1	3	5	4	7
P_3	2	6	8	6	12
		Avg = 4.6	A = 4.3	A = 3.3	Avg = 8

Grant chart	P_1	P_2	P_3	
exit time	0	5	8	14

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.



(2) shoutest job first.

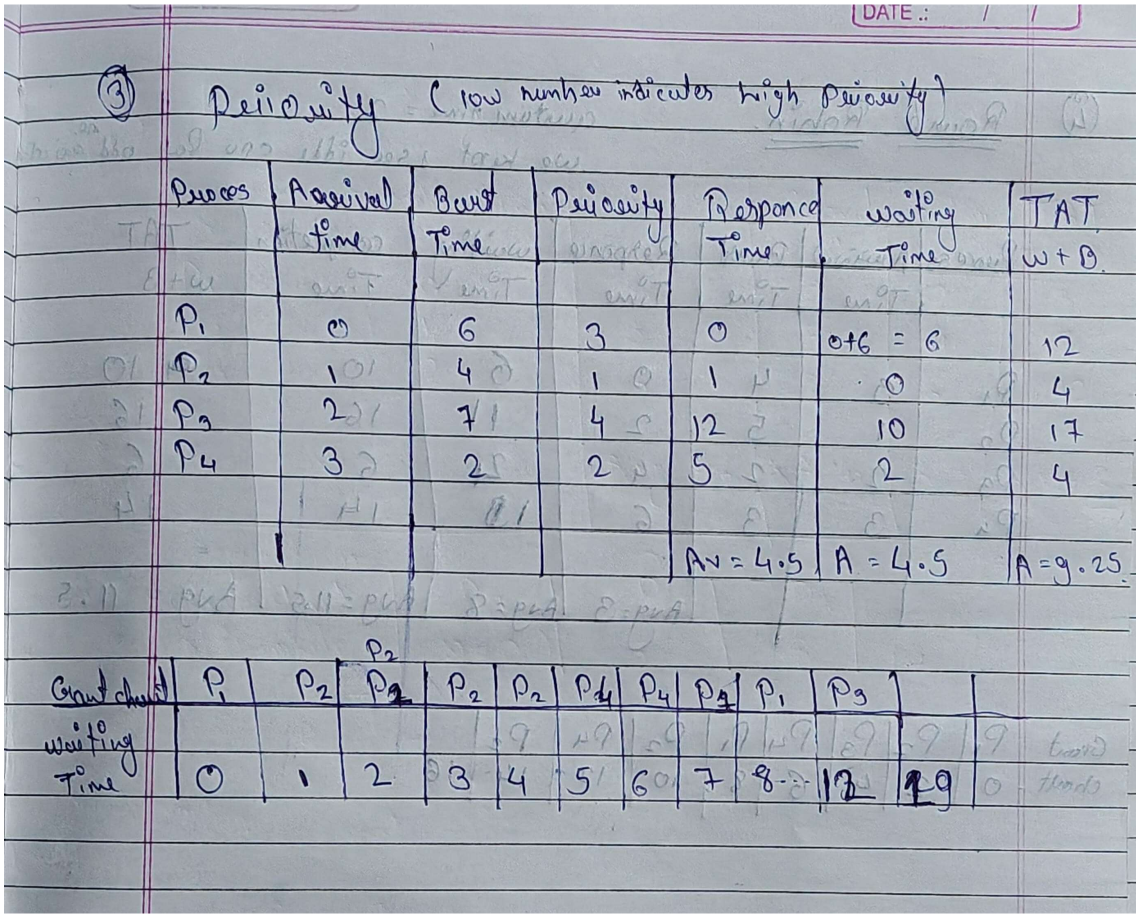
Process	Arrival time	Burst Time	Responce time	waiting Time	TAT w+B
P1	0	3	0	1	4
P2	1	5	9	9	14
P3	2	1	2	~~216~~ 2	3
P4	3	4	5	5	9
			Avg= 4	Avg = 4.25	Avg = 7.5

Gantt chart	P1	P1	P3	P1	P1	P4	P4	P4	P4	P2	P2	P2	P2	P2	
waiting	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

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.



DATE .: / /

③ Priority (low number indicates high priority)

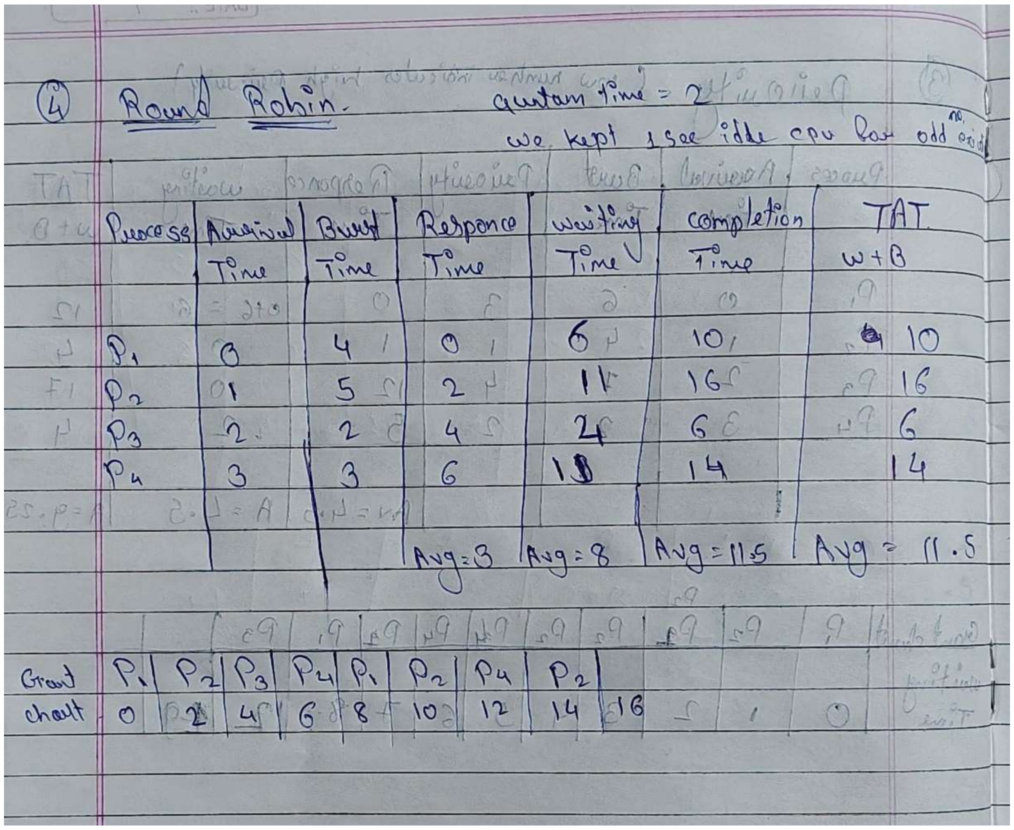
Proces	Arrival time	Burst Time	Priority	Responce Time	waiting Time	TAT W + B
P1	0	6	3	0	0+6 = 6	12
P2	1	4	1	1	0	4
P3	2	7	4	12	10	17
P4	3	2	2	5	2	4
				Av = 4.5	A = 4.5	A = 9.25

Gant chart	P1	P2	P2	P2	P2	P4	P4	P1	P1	P3		
waiting Time	0	1	2	3	4	5	6	7	8	12	19	

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.



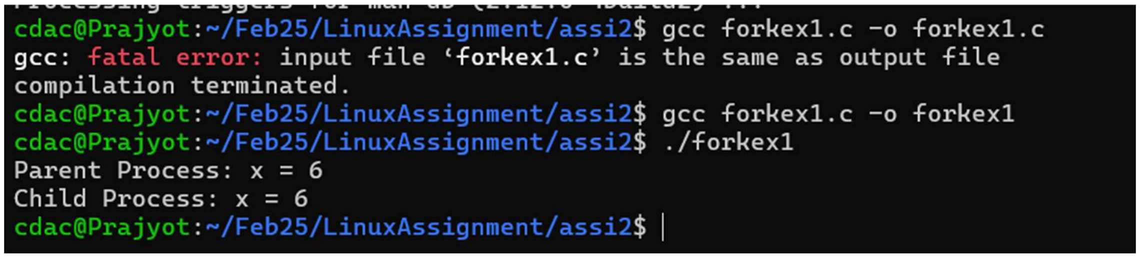
④ Round Robin. quntam time = 2

we kept 1 sec idle cpu for odd no. [illegible]

Process	Arrival Time	Burst Time	Responce Time	waiting Time	completion Time	TAT w + B
P_1	0	4	0	6	10	10
P_2	1	5	2	11	16	16
P_3	2	2	4	4	6	6
P_4	3	3	6	11	14	14
			Avg = 3	Avg = 8	Avg = 11.5	Avg = 11.5

Grant 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

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?



```
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ gcc forkex1.c -o forkex1.c
gcc: fatal error: input file ‘forkex1.c’ is the same as output file
compilation terminated.
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ gcc forkex1.c -o forkex1
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ ./forkex1
Parent Process: x = 6
Child Process: x = 6
cdac@Prajyot:~/Feb25/LinuxAssignment/assi2$ |
```