Assignment 2

Part A

What will the following commands do?

* echo "Hello, World!"

= It prints the text under “ ” with variables ,characters, etc.

* name="Productive"

= It display the name.

* touch file.txt

= It create new file.

* ls -a

= It displays files and directories .with bash history ,cache, and logout ,etc.

* rm file.txt

= It removes the file from directory.

* cp file1.txt file2.txt

= copy the file to source file to the destination file. File1 is the source file and file2 is the destination file.

* mv file.txt /path/to/directory/

= moves files and directories from one directory to another or renames a file or directory.

* chmod 755 script.sh

1. = It gives the permission to read write and execute. chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.

* grep "pattern" file.txt

= It perform text searches for a defined criteria of words or string.

* kill PID

= To kill a process from the linux command line.

* mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt  ls -l | grep ".txt"

= It will create a directory named “mydir” ,in this directory create a new file named “file.txt” and prints hello world by using eco command ,and find the text as “txt”.

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

= concatenate the file1.txt and file2.txt and sort the text.

* ls -l | grep "^d"

= ls -l it display the list of files and

cdac@Shrutika:~$ ls -l |grep "^d"

drwxr-xr-x 2 cdac cdac 4096 Aug 28 21:29 LinuxAssignment

drwxr-xr-x 3 cdac cdac 4096 Aug 28 22:09 New

drwxr-xr-x 2 cdac cdac 4096 Aug 28 21:33 docs

drwxr-xr-x 2 cdac cdac 4096 Aug 31 14:25 mydir

* grep -r "pattern" /path/to/directory/

= it search the text in to the directory.

* cat file1.txt file2.txt | sort | uniq –d
* chmod 644 file.txt
* cp -r source\_directory destination\_directory

= it copies the file from source file to destination file

* find /path/to/search -name "\*.txt"

* chmod u+x file.txt

= it gives permission to file. to user the execute text

* echo $PATH

= cdac@Shrutika:~$ 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/Program Files/Common Files/Oracle/Java/javapath:/mnt/c/Windows/system32:/mnt/c/Windows:/mnt/c/Windows/System32/Wbem:/mnt/c/Windows/System32/WindowsPowerShell/v1.0/:/mnt/c/Windows/System32/OpenSSH/:/mnt/c/Program Files/dotnet/:/mnt/c/Program Files/Git/cmd:/mnt/c/Users/s9037/AppData/Local/Microsoft/WindowsApps:/mnt/c/Users/s9037/AppData/Local/Programs/Microsoft VS Code/bin:/snap/bin

# Part B

Identify True or False:

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

=true

1. mv is used to move files and directories.

=true

1. cd is used to copy files and directories.

=false

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

=true

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

=true

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

=true

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

=false

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

=true

Identify the Incorrect Commands:

1. chmodx is used to change file permissions.

=chmod

1. cpy is used to copy files and directories.

=cp

1. mkfile is used to create a new file.

=touch is used to create new file

1. catx is used to concatenate files.

=cat is used to concatenate files.

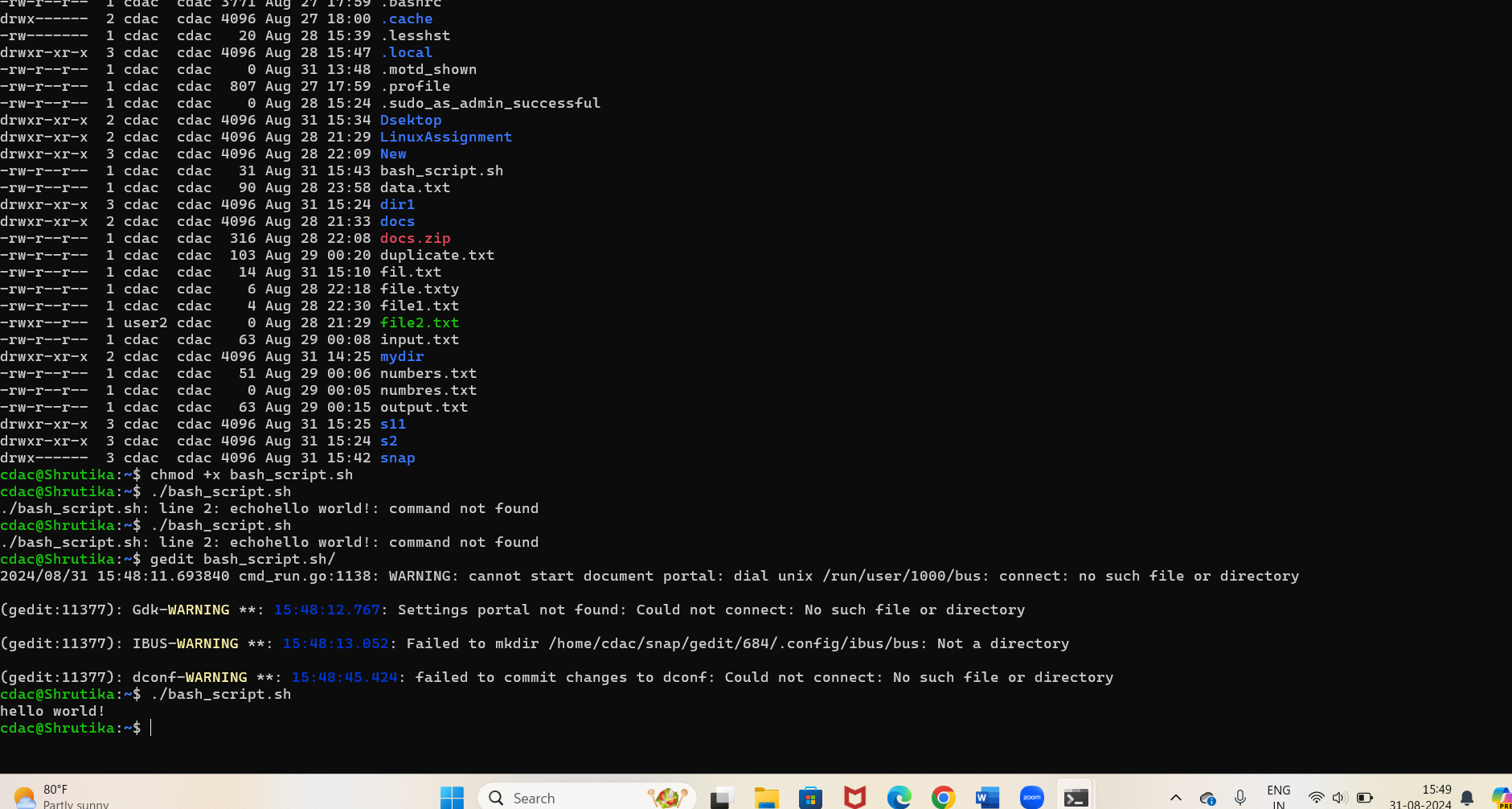
1. rn is used to rename files.

=mv is used to rename the files.

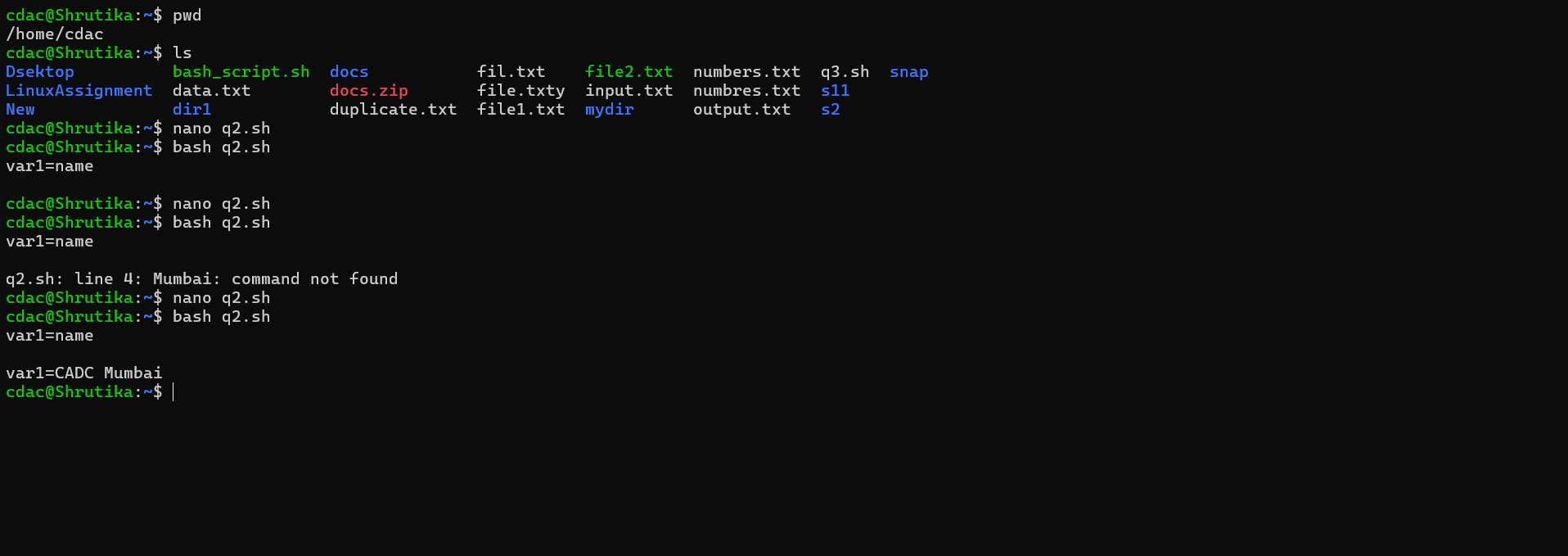
# Part C

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

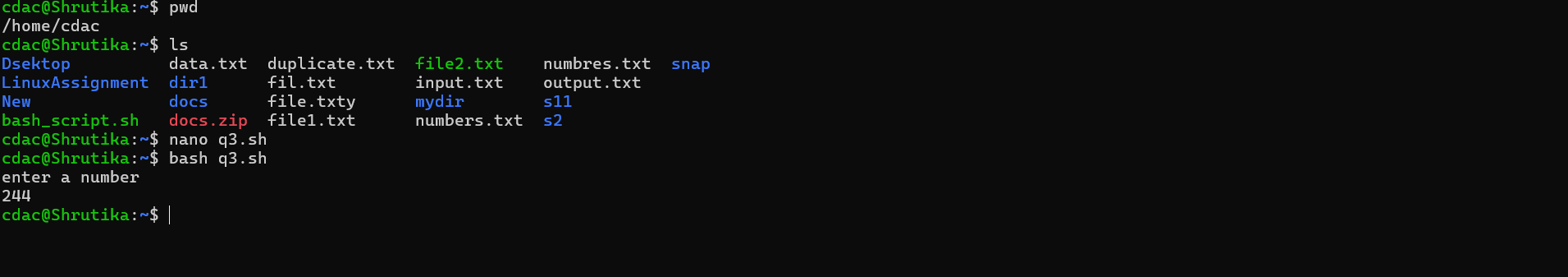
Ans-



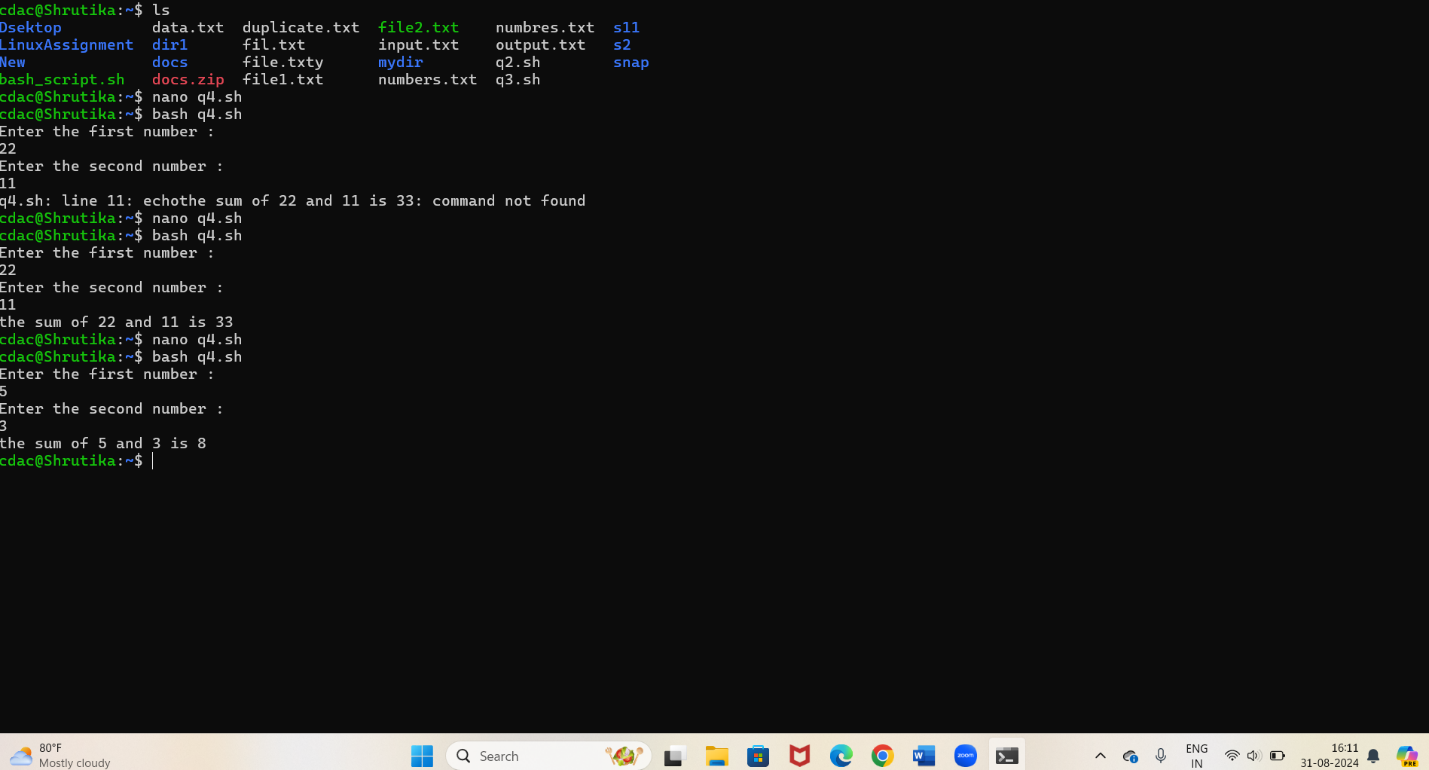
Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.



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

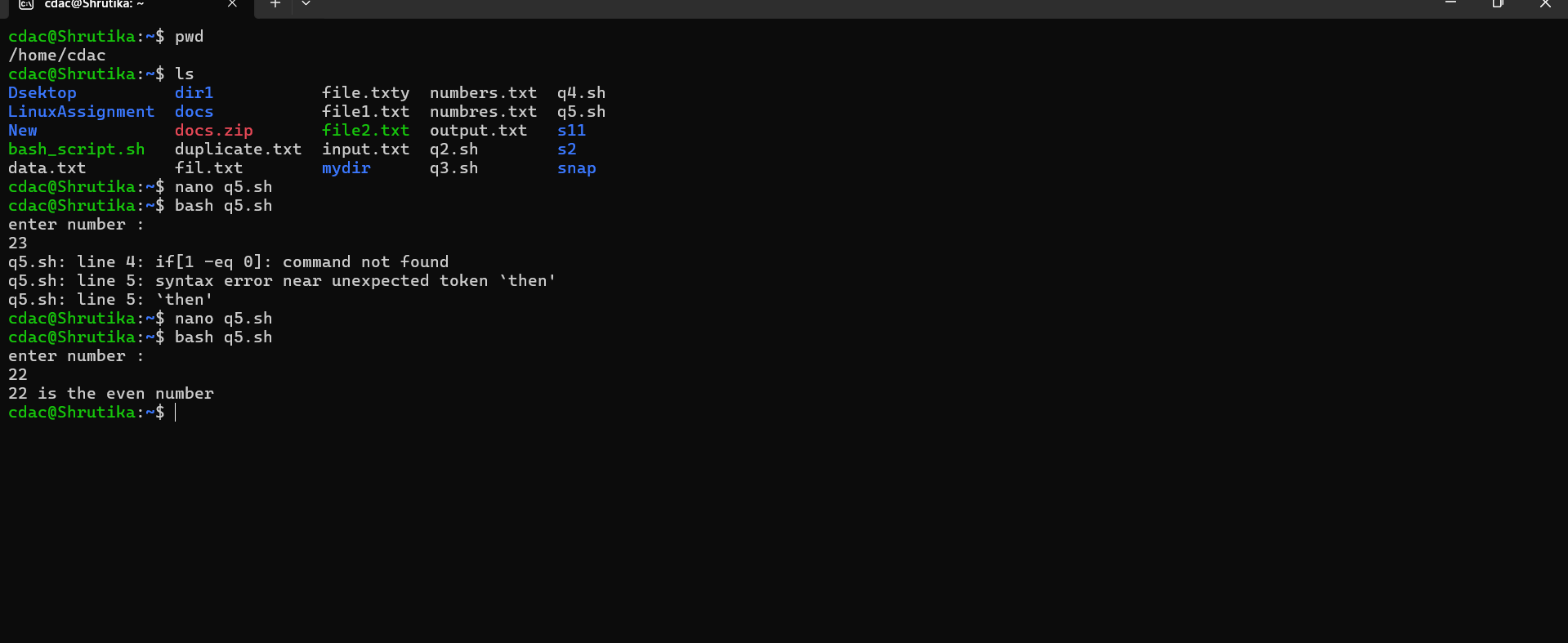


Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result

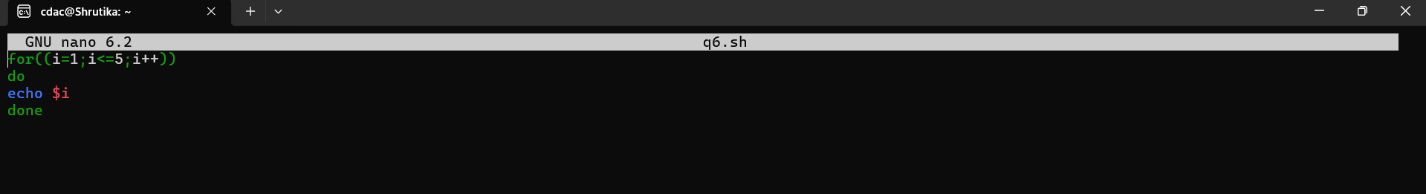


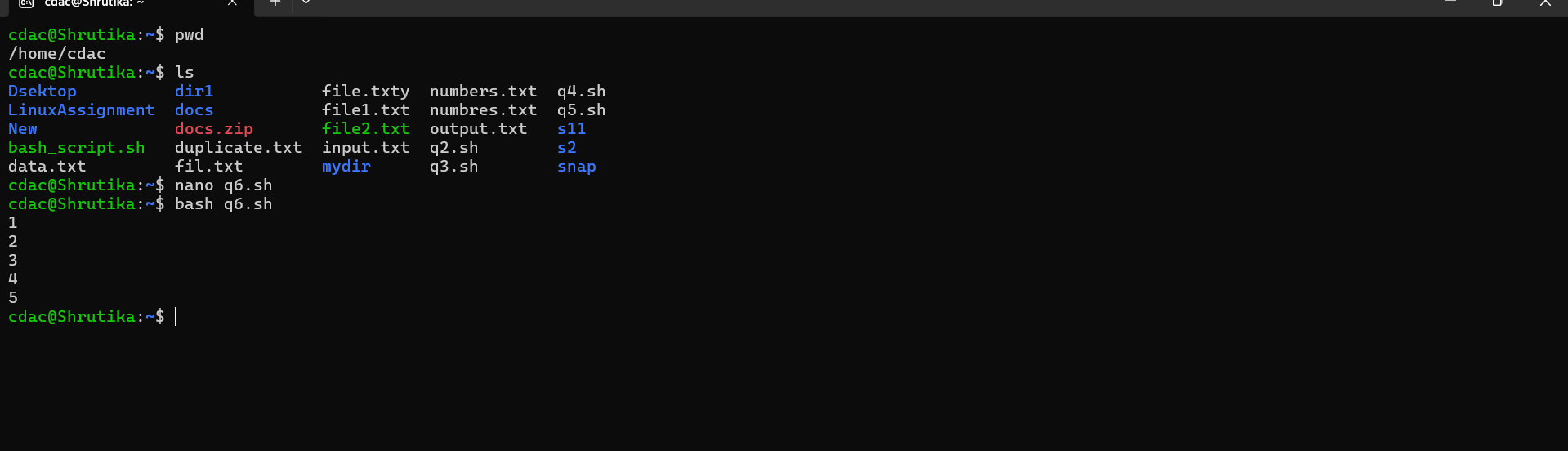
.

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

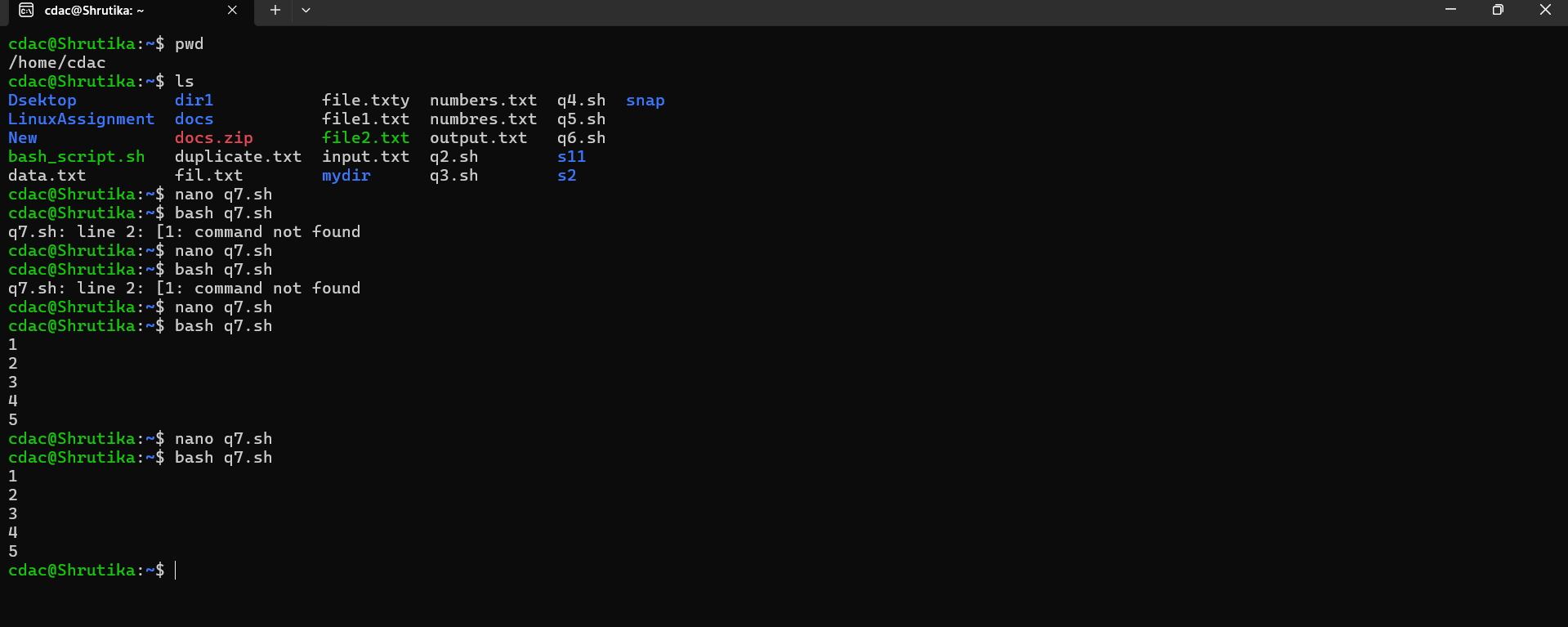


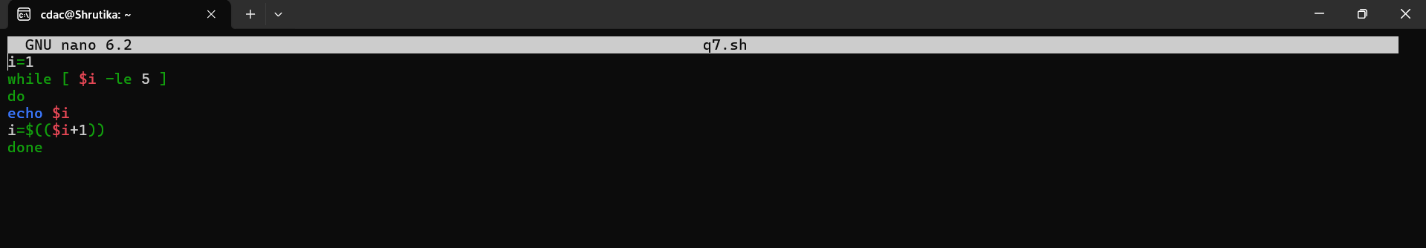
Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.



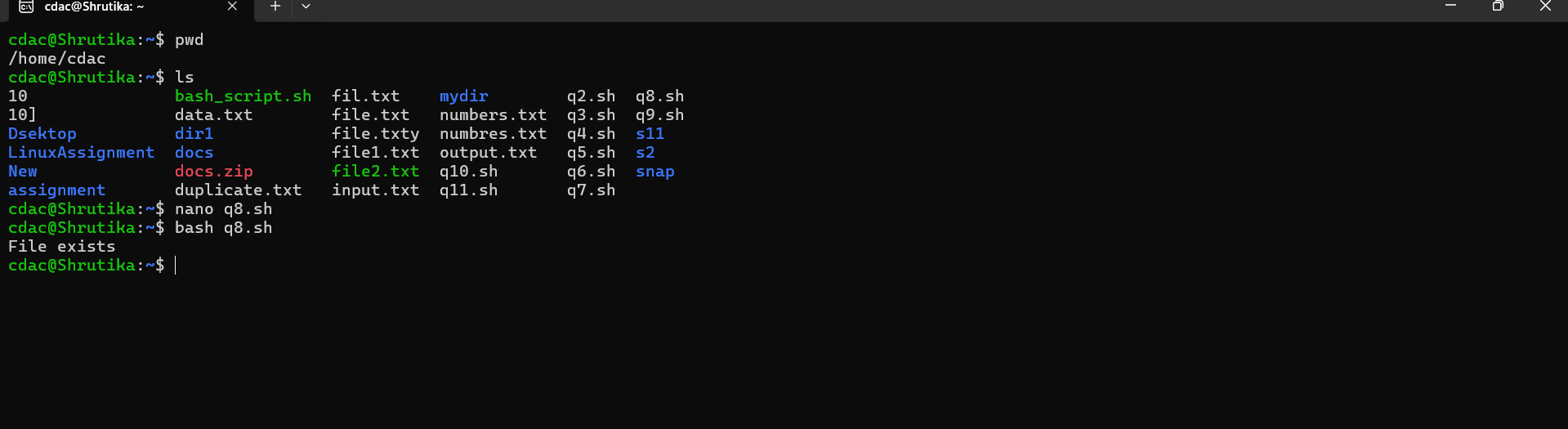


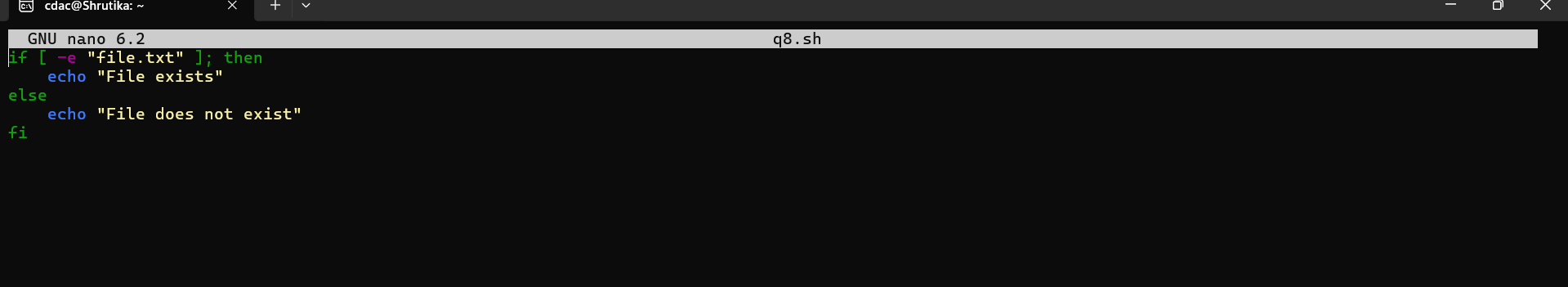
Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.



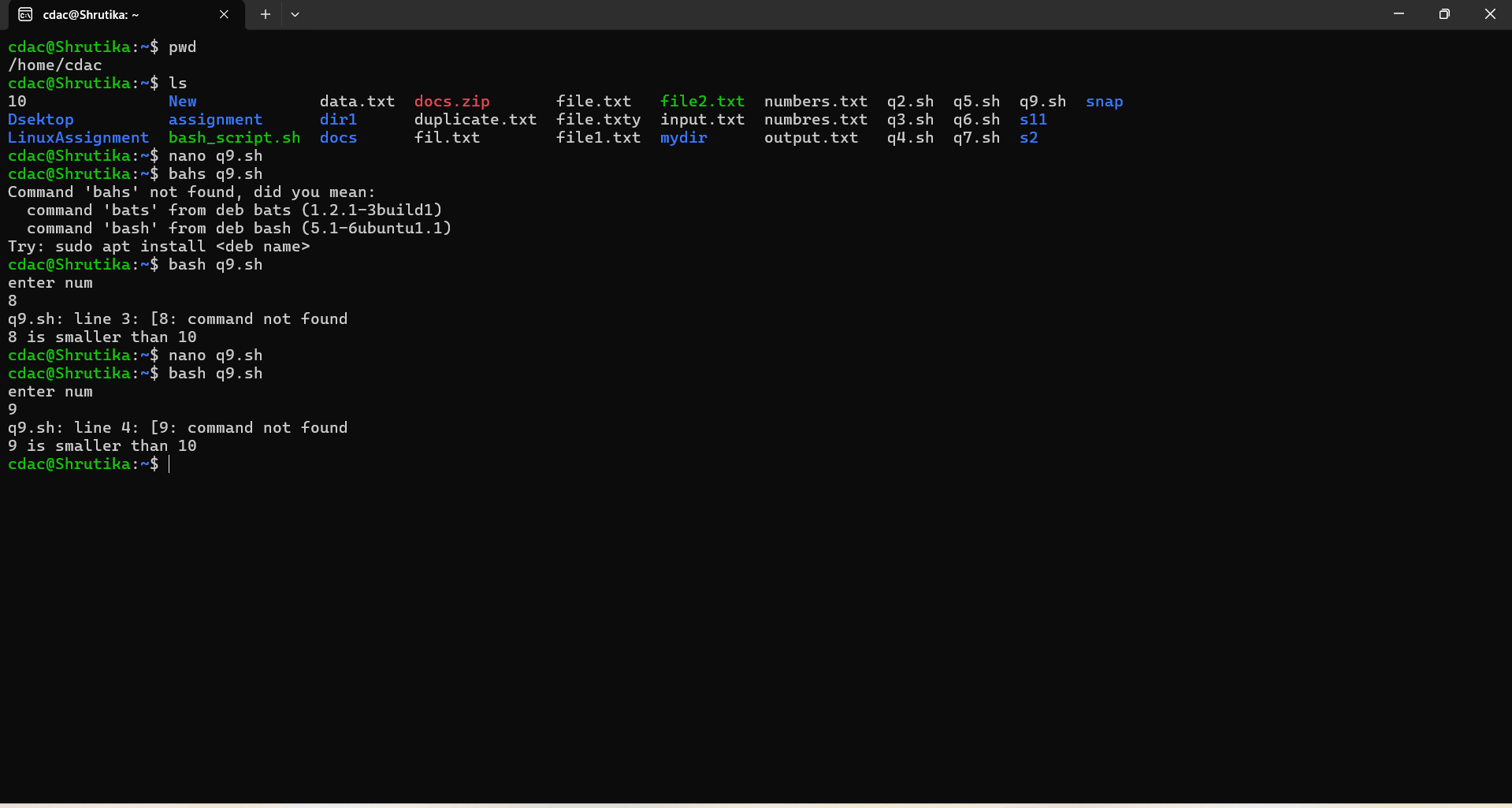


Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".

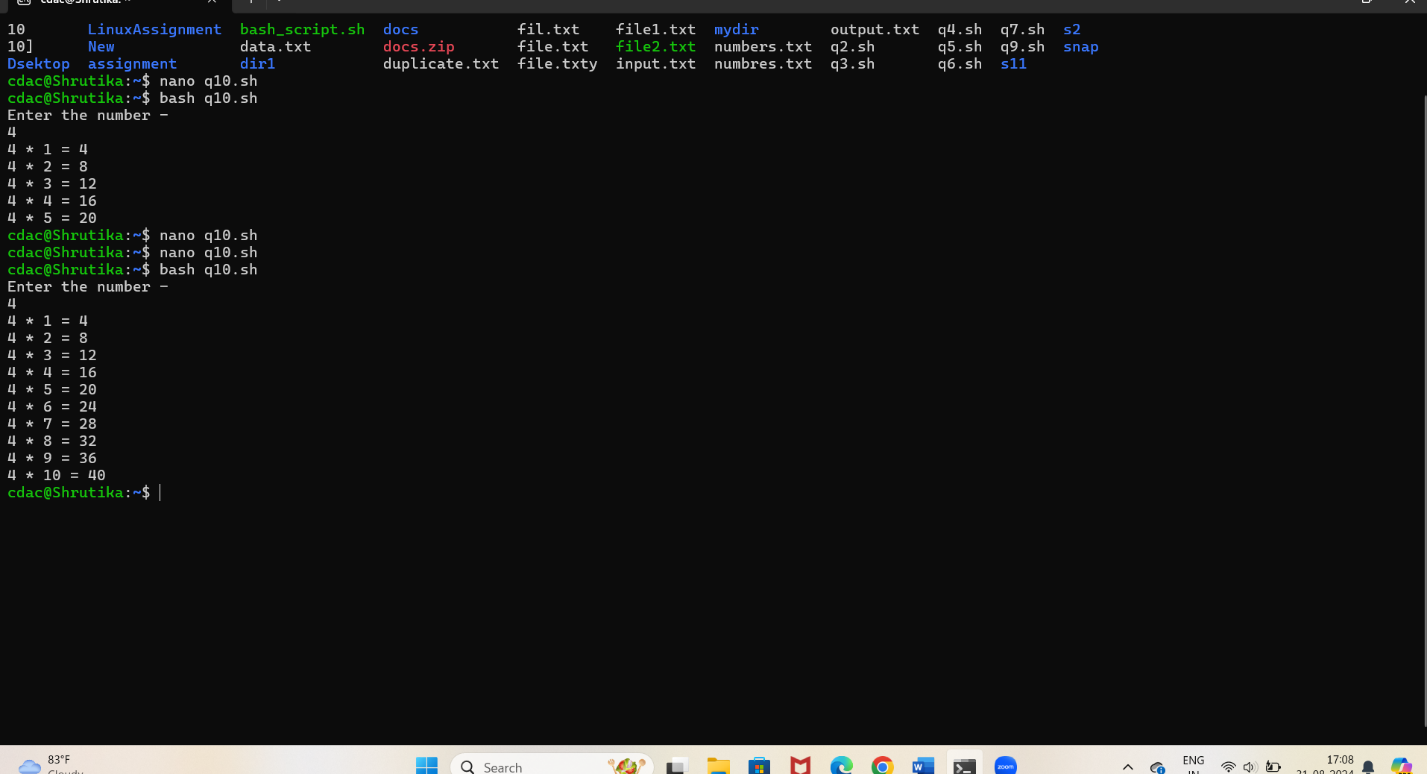




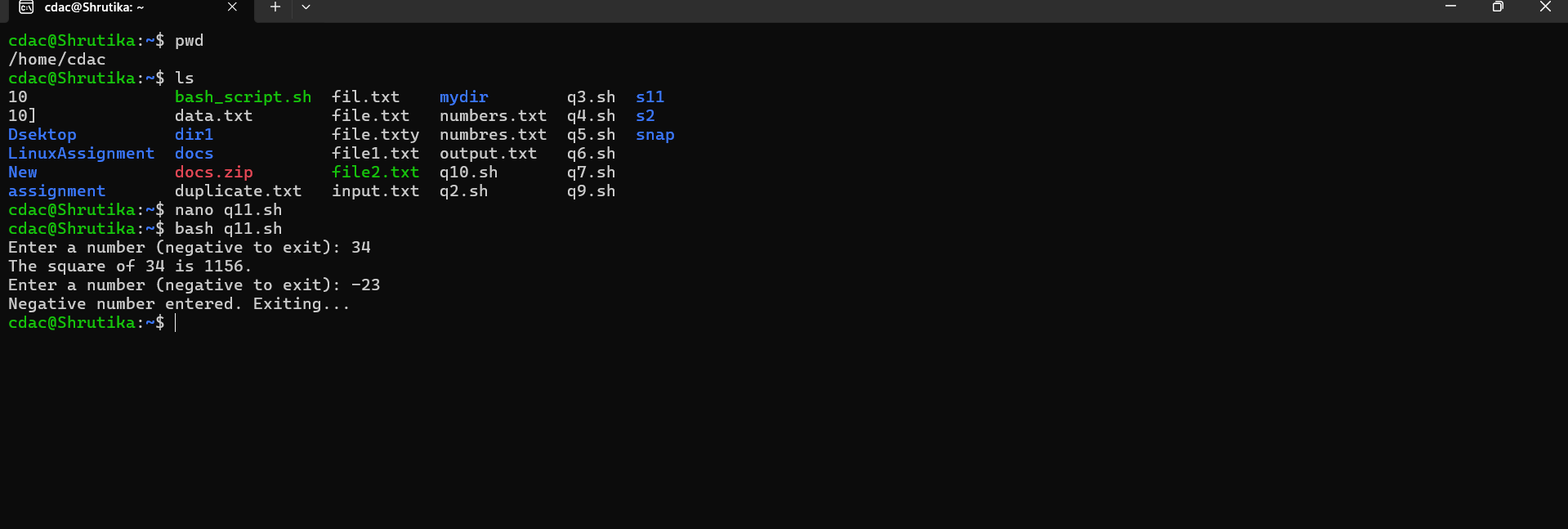
Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.



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 |

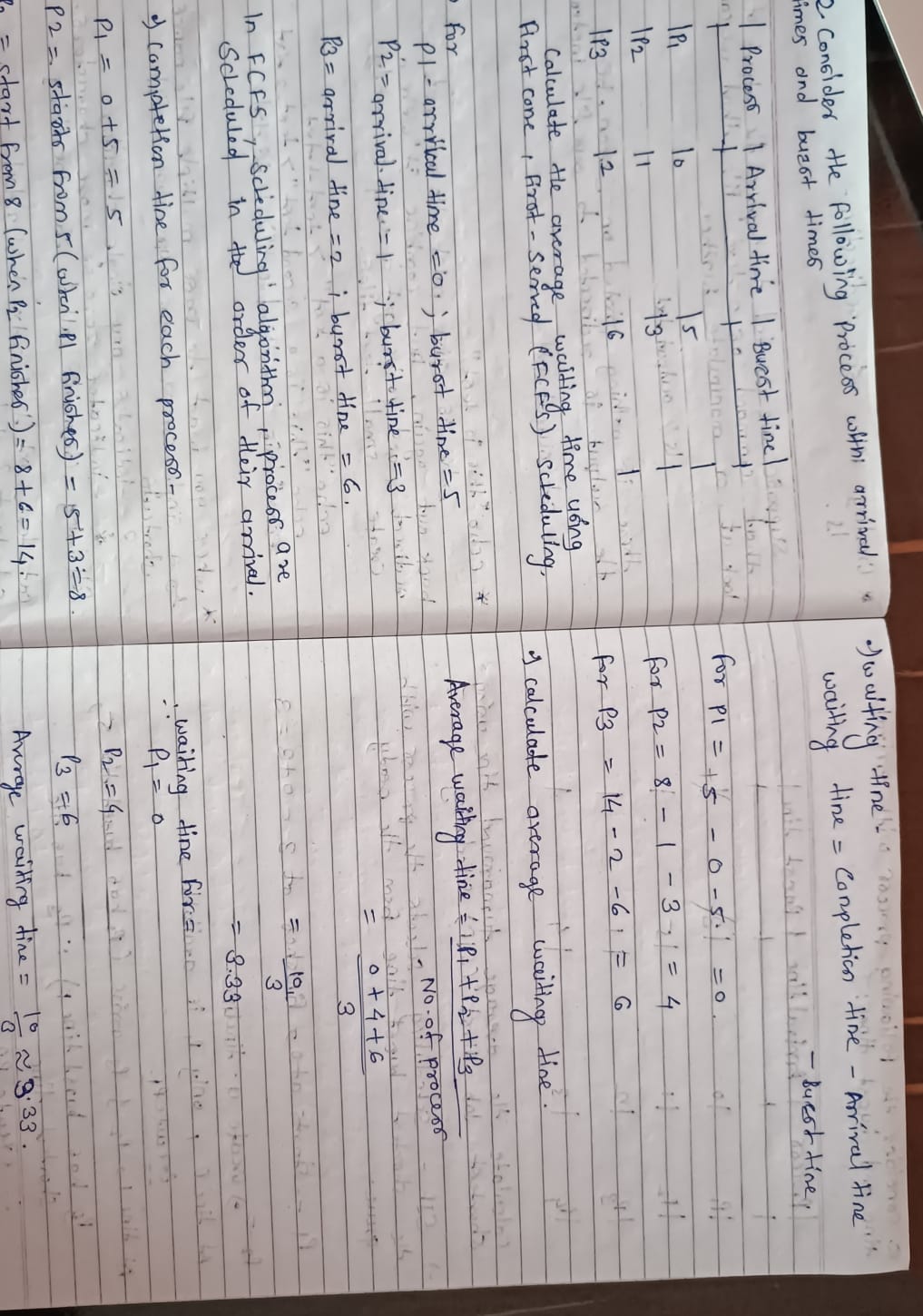
|---------|--------------|------------|

| P1 | 0 | 5 |

| P2 | 1 | 3 |

| P3 | 2 | 6 |

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



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

| Process | Arrival Time | Burst Time |

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

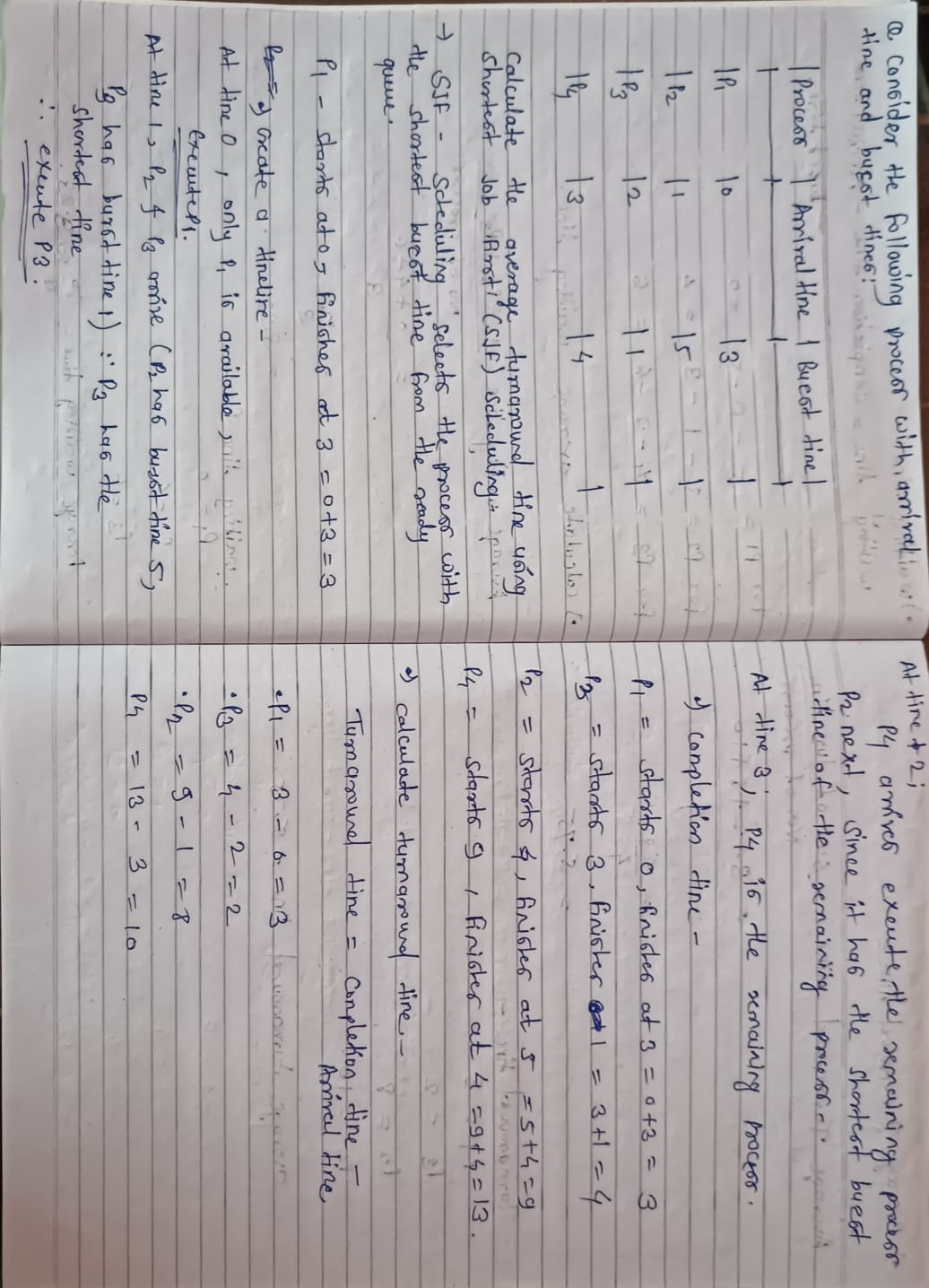
| P1 | 0 | 3 |

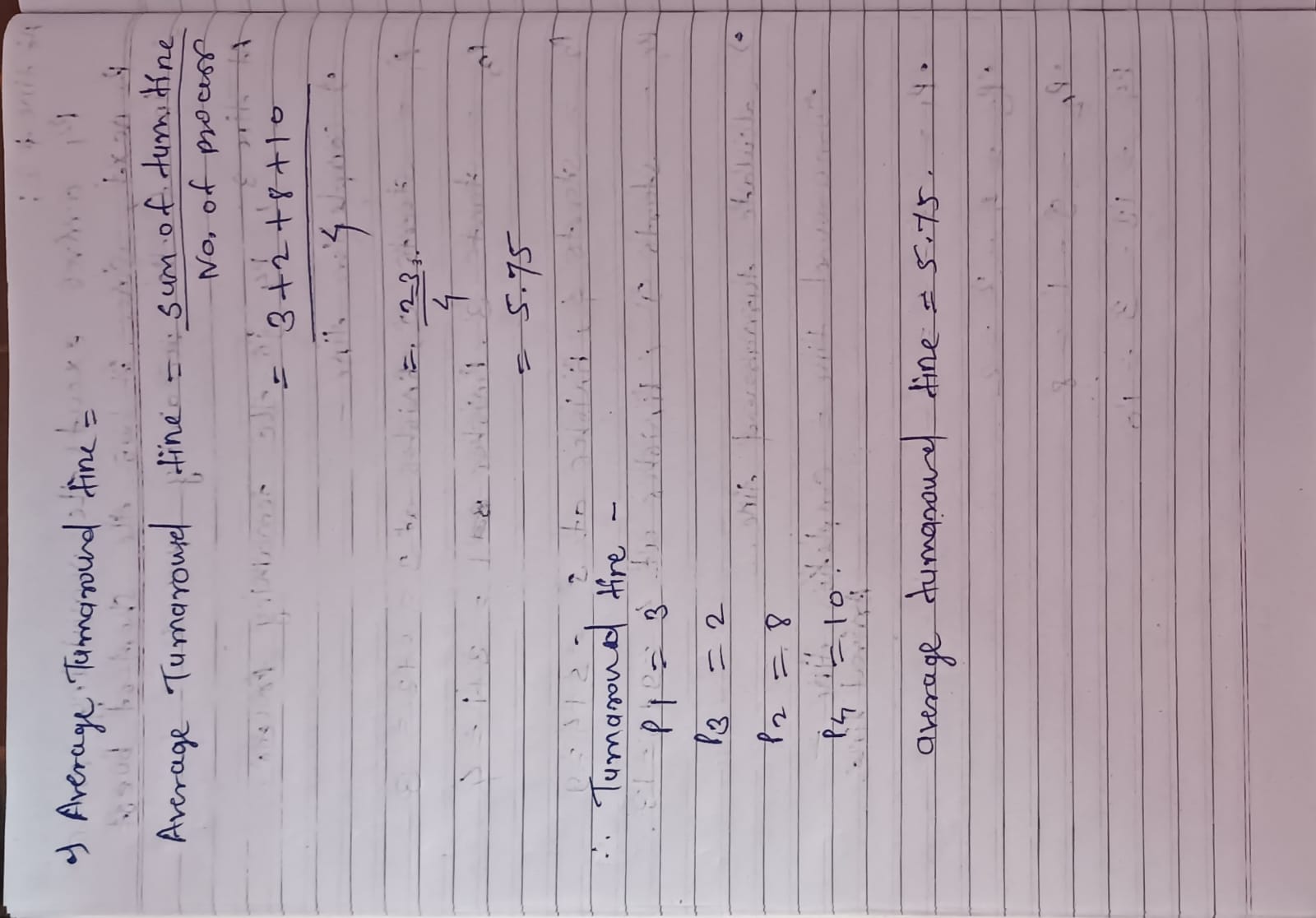
| P2 | 1 | 5 |

| P3 | 2 | 1 |

| P4 | 3 | 4 |

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

eduling.



1. Consider the following processes with arrival times, burst times, and priorities (lower number indicates higher priority):

| Process | Arrival Time | Burst Time | Priority |

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

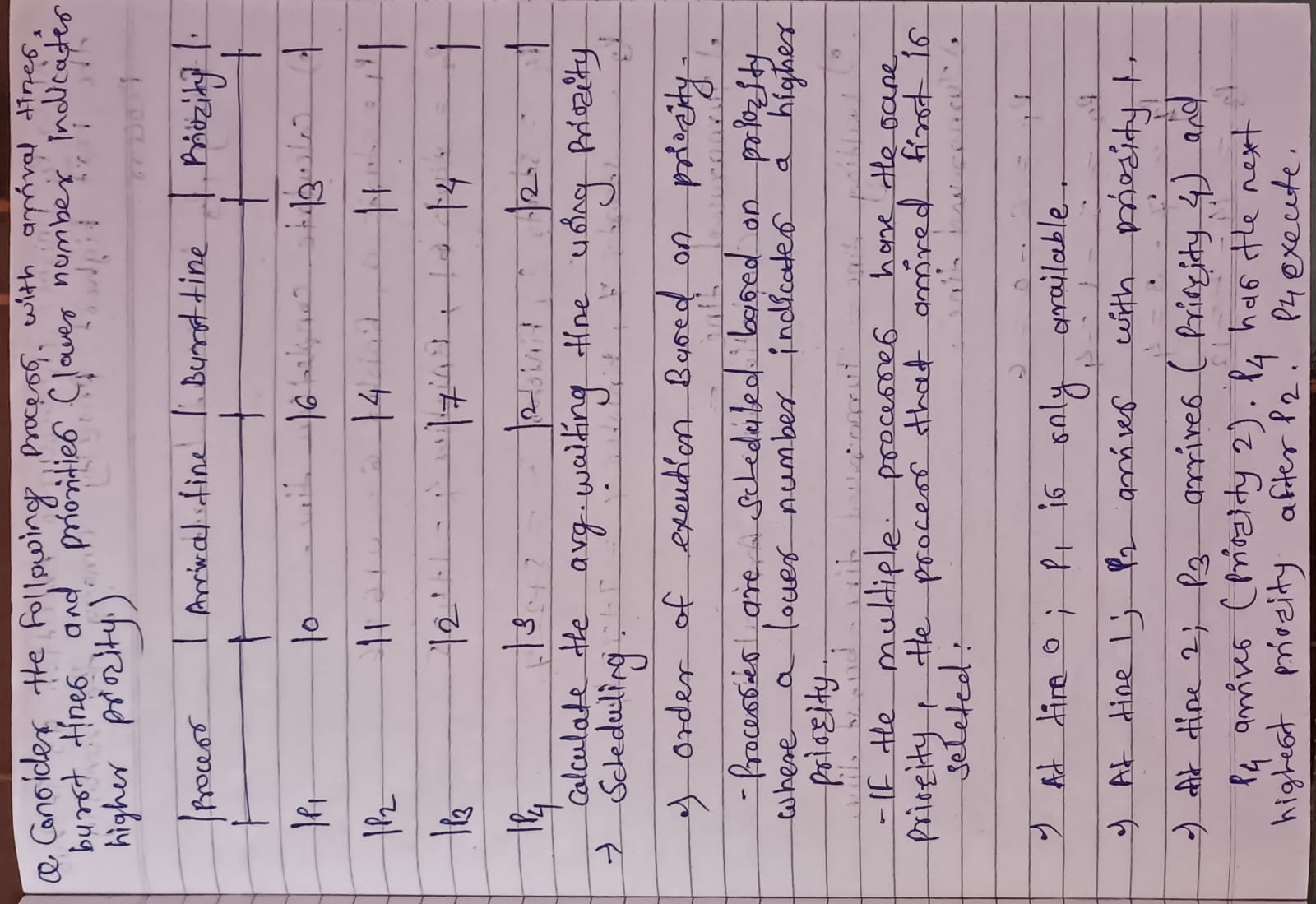
| P1 | 0 | 6 | 3 |

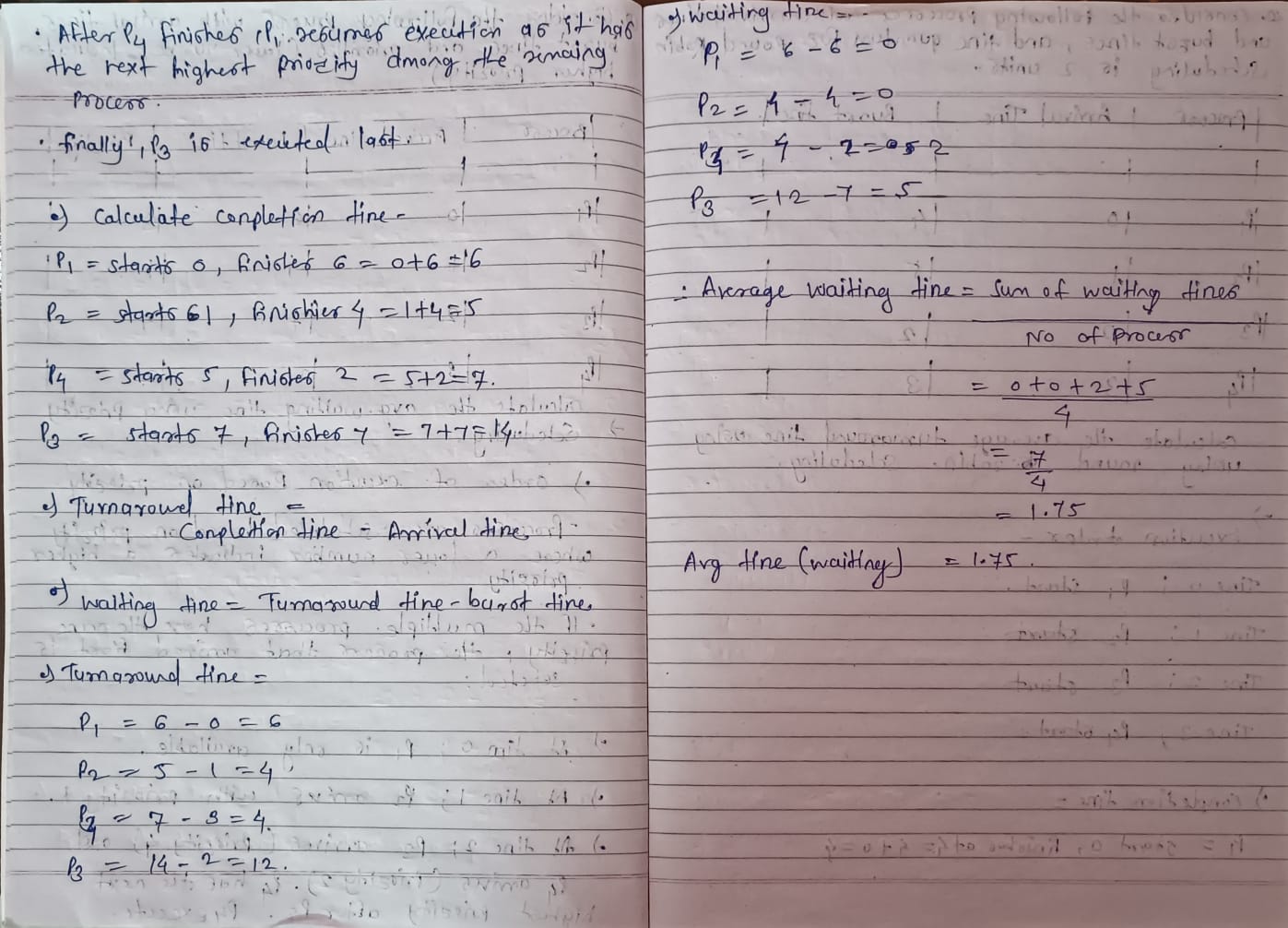
| P2 | 1 | 4 | 1 |

| P3 | 2 | 7 | 4 |

| P4 | 3 | 2 | 2 |

Calculate the average waiting time using Priority Scheduling.





1. 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 |

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

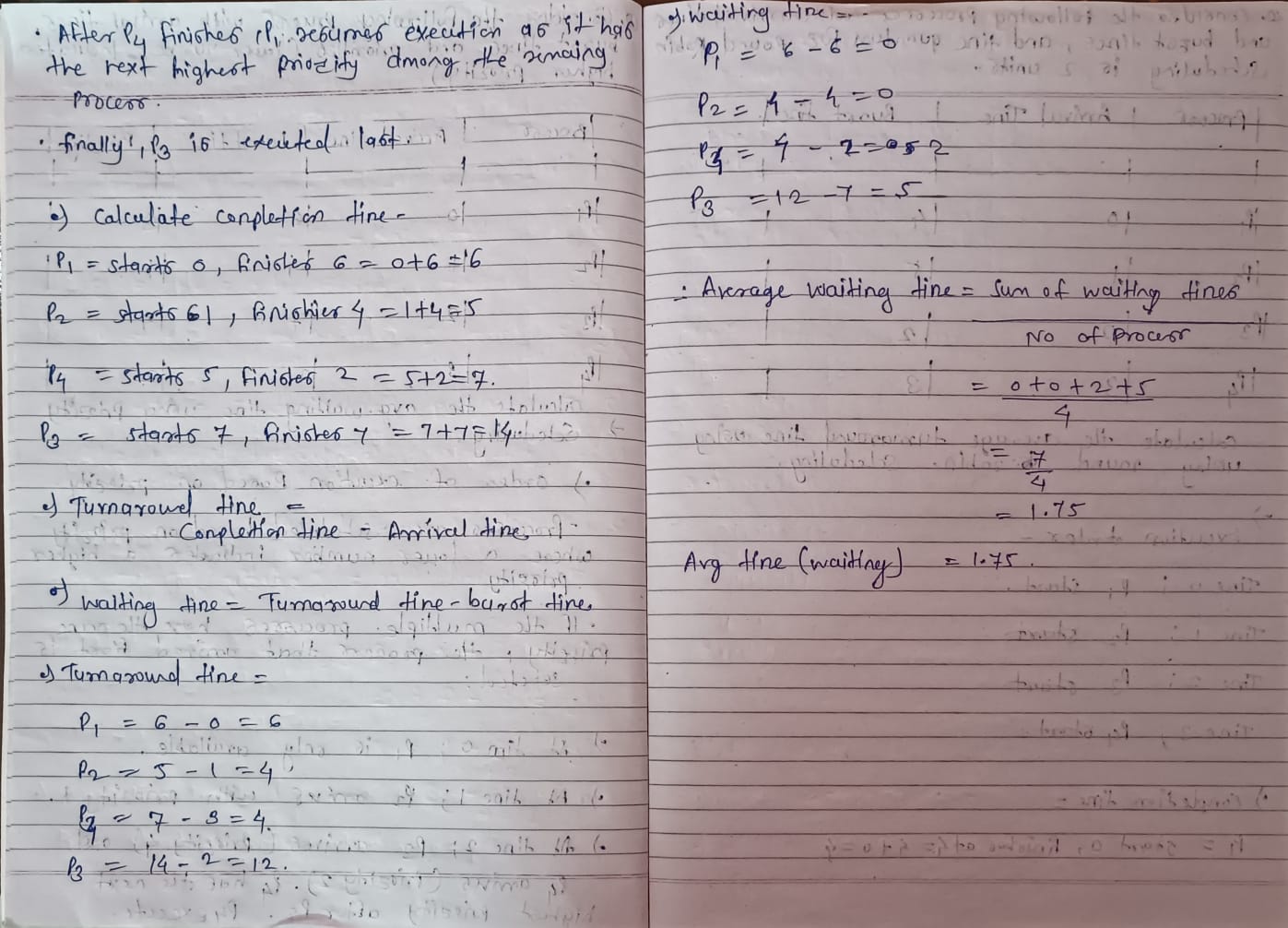
| P1 | 0 | 4 |

| P2 | 1 | 5 |

| P3 | 2 | 2 |

| P4 | 3 | 3 |

Calculate the average turnaround time using Round Robin scheduling.



1. 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?

