**Part A**

**What will the following commands do?**

* **echo "Hello, World!":** Prints "Hello, World!" to the terminal.
* **name="Productive":** Assigns the value "Productive" to the variable name.
* **touch file.txt:** Creates an file named file.txt.
* **ls -a:** Lists all files, including hidden ones, in the directory.
* **rm file.txt:** Deletes the file file.txt.
* **cp file1.txt file2.txt:** Copies file1.txt to file2.txt.
* **mv file.txt /path/to/directory/:** Moves file.txt to the specified directory.
* **chmod 755 script.sh:** Gives the owner full permissions and others read/execute on script.sh.
* **grep "pattern" file.txt:** Searches for the word "pattern" in file.txt.
* **kill PID:** Terminates the process with the given PID.
* **mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt:** Creates directory mydir, navigates to it, creates file.txt, writes "Hello, World!" into it, and displays the content.
* **ls -l | grep ".txt":** Lists files in long format and shows only .txt files.
* **cat file1.txt file2.txt | sort | uniq**: Combines content of file1.txt and file2.txt, sorts them, and removes duplicate lines.
* **ls -l | grep "^d":** Lists directories in long format.
* **grep -r "pattern" /path/to/directory/:** Recursively searches for "pattern" in all files within the directory.
* **cat file1.txt file2.txt | sort | uniq -d**: Shows only the lines that are duplicated in both file1.txt and file2.txt.
* **chmod 644 file.txt:** Gives the owner read/write permissions and others read-only on file.txt.
* **cp -r source\_directory destination\_directory:** Copies the entire source\_directory to destination\_directory.
* **find /path/to/search -name "\*.txt":** Finds all .txt files in the specified directory.
* **chmod u+x file.txt:** Gives execute permission to the owner on file.txt.
* **echo $PATH:** Displays any value assigned to PATH variable.

**True/False:**

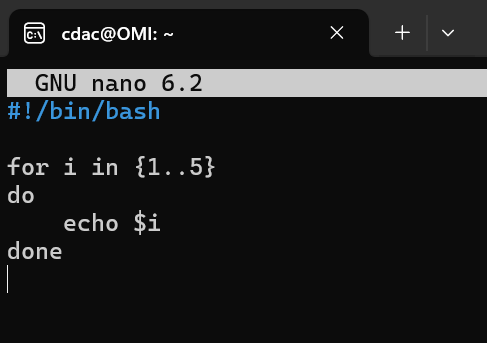
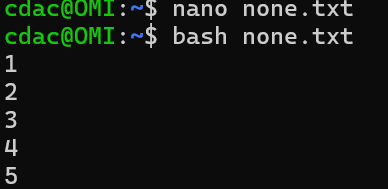
1. **True** - ls lists files and directories.
2. **True** - mv moves files and directories.
3. **False** - cd is used to change directories, not copy.
4. **True** - pwd shows the current directory.
5. **True** - grep searches for patterns in files.
6. **True** - chmod 755 gives the mentioned permissions.
7. **True** - mkdir -p creates nested directories.
8. **True** - rm -rf deletes files forcefully without asking.

**Incorrect Commands:**

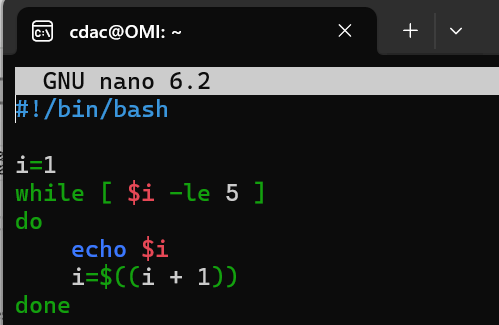
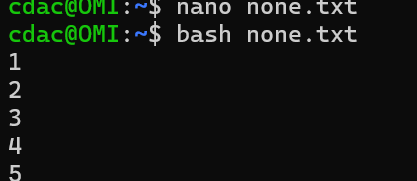
1. **False** - chmodx command is not used to change file permission, it's chmod.
2. **False** - cpy command is not for copying files, it's cp.
3. **False** - mkfile command is not used to create a file, it's touch.
4. **False** - catx command is not used for concatenating, it's cat.
5. **False** - rn command is not for renaming files, it's mv.

PART-C

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

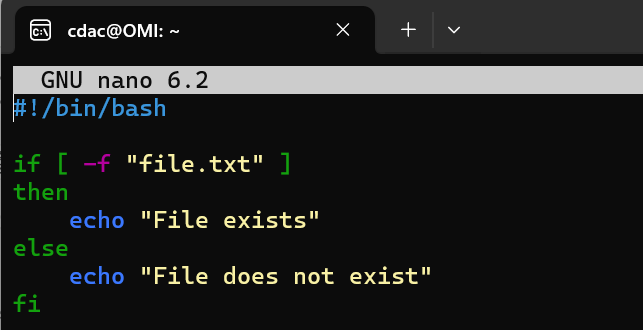
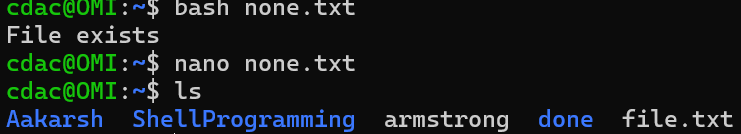
INPUT : OUTPUT: 

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

 OUTPUT: 

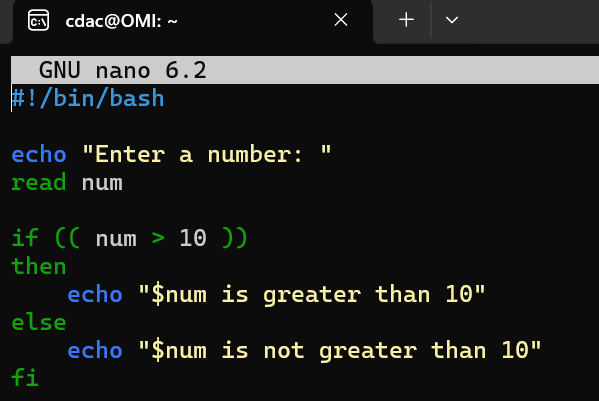
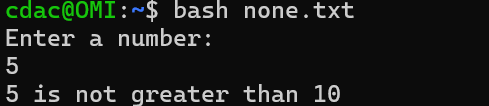
Question : Write a shell script that checks if a file named "file.txt" exists in the current directory. If it

does, print "File exists", otherwise, print "File does not exist".

 *OUTPUT:* 

Question: Write a shell script that uses the if statement to check if a number is greater than 10 and

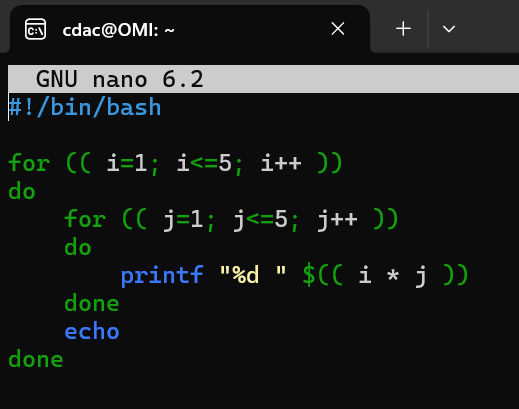
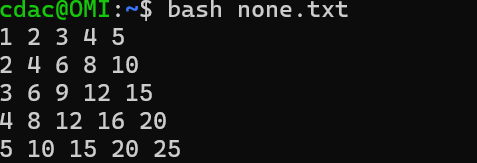
prints a message accordingly.

 OUTPUT: 

Question: 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.

 *OUTPUT:* 

Part E

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

| Process | Arrival Time | Burst Time |

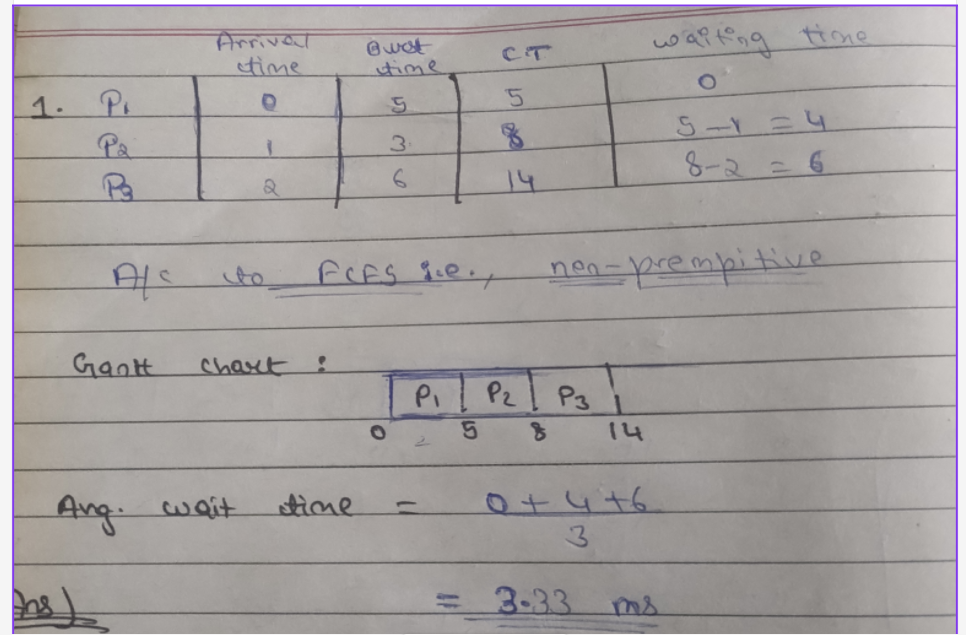
|---------|--------------|------------|

| P1 | 0 | 5 |

| P2 | 1 | 3 |

| P3 | 2 | 6 |

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



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

| Process | Arrival Time | Burst Time |

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

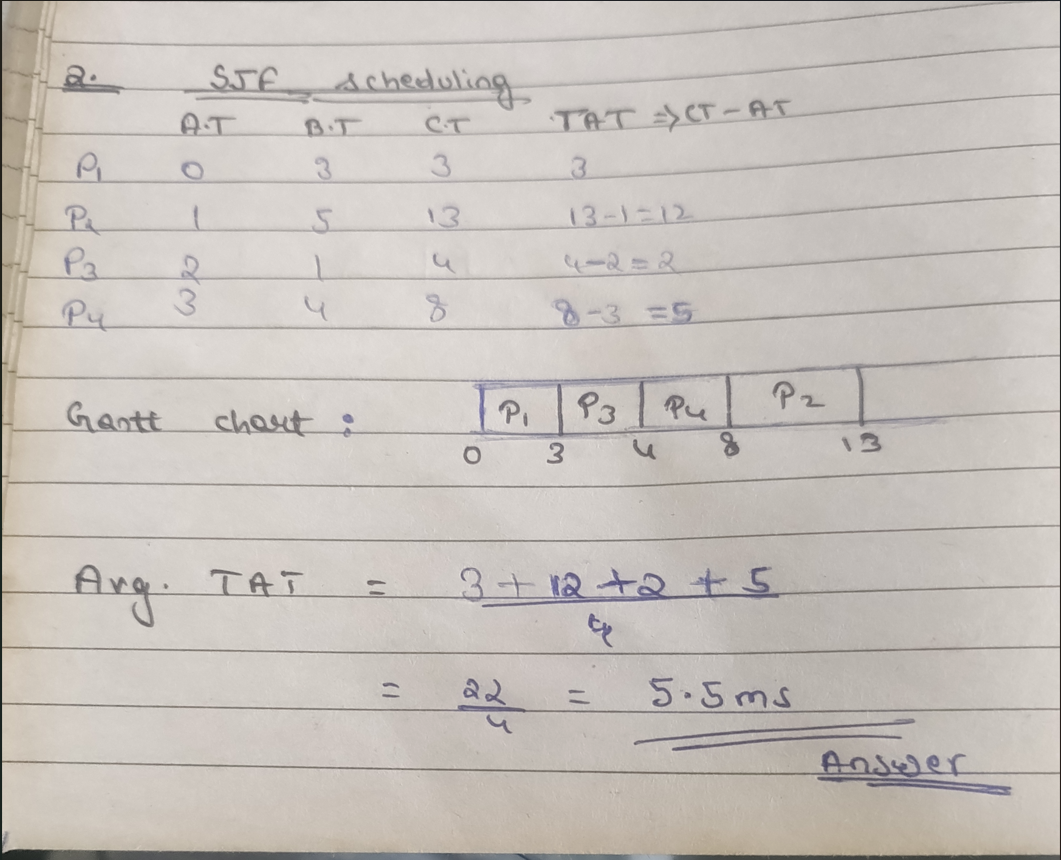
| P1 | 0 | 3 |

| P2 | 1 | 5 |

| P3 | 2 | 1 |

| P4 | 3 | 4 |

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



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

indicates higher priority):

| Process | Arrival Time | Burst Time | Priority |

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

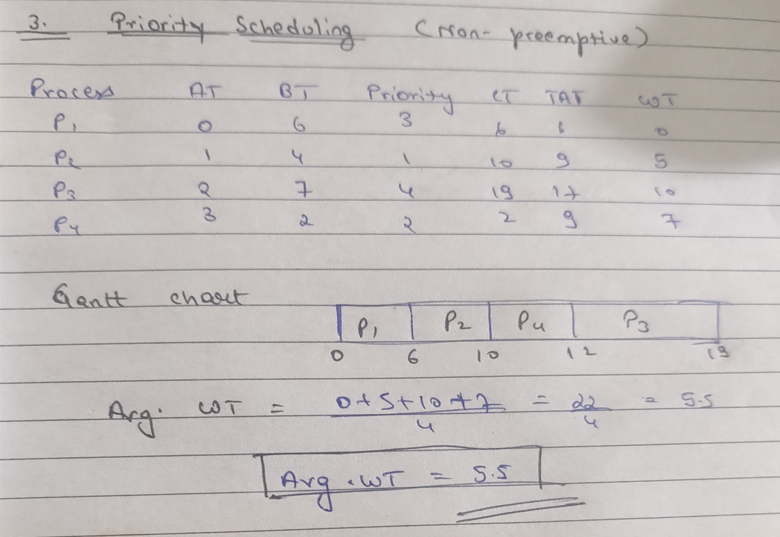
| P1 | 0 | 6 | 3 |

| P2 | 1 | 4 | 1 |

| P3 | 2 | 7 | 4 |

| P4 | 3 | 2 | 2 |

Calculate the average waiting time using Priority Scheduling.



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.

