**CDAC MUMBAI**

**Concepts of Operating System**

**Assignment 2**

**Part A**

What will the following commands do?

• echo "Hello, World!"

* This command prints the string "Hello, World!" to the terminal. The echo command is used to output text to the terminal.

• name="Productive"

* This command sets a variable named name to the value "Productive". Variables are used to store values that can be used later in a program.

• touch file.txt

* touch command Creates an empty file named file.txt if it doesn’t already exist.

• ls -a

* Lists all files and directories in the current directory, including hidden files.

• rm file.txt

* rm command used to Delete the file named file.txt.

• cp file1.txt file2.txt

* Copies the contents of file1.txt into a new file named file2.txt.

• mv file.txt /path/to/directory/

* This command moves the file file.txt to the specified directory. The mv command is used to move or rename files and directories.

• chmod 755 script.sh

* Changes the permissions of the file script.sh to 755, making it readable and executable by everyone, but writable only by the owner.

• grep "pattern" file.txt

* Searches for the string "pattern" in file.txt and displays all lines containing that pattern.

• kill PID

* The kill command is used to terminate processes.

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

* Creates a directory named mydir, navigates into it, creates a file named file.txt, writes "Hello, World!" into the file, and then displays the content of file.txt.

• ls -l | grep ".txt"

* This command lists all files and directories in the current directory in a long format (ls -l), and then pipes the output to grep, which searches for lines that contain ".txt".

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

* This command lists all files and directories in the current directory in a long format (ls -l), and then pipes the output to grep, which searches for lines that contain ".txt".

• ls -l | grep "^d"

* Lists files in long format (ls -l), then filters and displays only directories lines starting with "d".

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

* This command searches for the pattern "pattern" in all files recursively in the specified directory and its subdirectories. The -r option stands for "recursive".

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

* This command concatenates the contents of file1.txt and file2.txt, sorts the output, and then removes duplicate lines, but only keeps the duplicate lines that appear more than once.

• chmod 644 file.txt

* This command sets the permissions of the file file.txt to rw-r--r-- (read and write permissions for the owner, read permissions for the group, and read permissions for others).

• cp -r source\_directory destination\_directory

* This command copies the entire source\_directory and its contents to the destination\_directory. The -r stands for "recursive".

• find /path/to/search -name "\*.txt"

* Searches for all files with a .txt extension in /path/to/search and its subdirectories.

• chmod u+x file.txt

* Adds execute permissions for the owner (user) of file.txt.

• echo $PATH

* Displays the contents of the PATH environment variable, which lists directories where the system looks for executable files.

**Part B**

**Q1 Identify True or False:**

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

- This statement is **True** ls is used to list files and directories.

2. mv is used to move files and directories.

- **True**

3. cd is used to copy files and directories.

- **False** Because cd command is used to change the current working directory.

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**

**Q2 Identify the Incorrect Commands:**

1. chmodx is used to change file permissions.

- chmodx is incorrect command “chmod” command is right to change file permissions.

2. cpy is used to copy files and directories.

- “cp” command is correct to copy files and directories.

3. mkfile is used to create a new file.

- To create a new file “touch” and “nano” command is used.

4. catx is used to concatenate files.

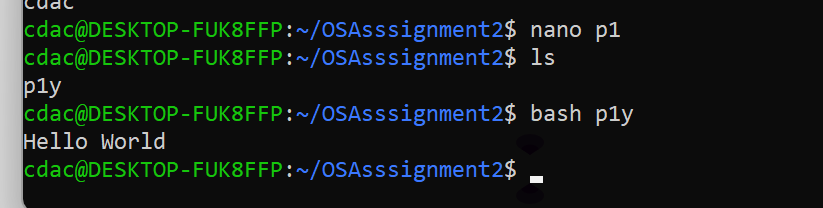
- “cat” command is used to concatenate file.

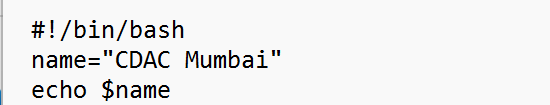
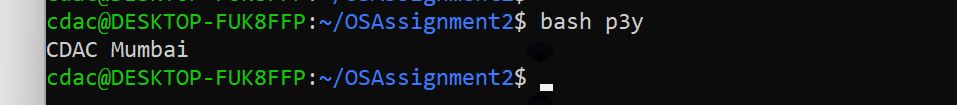
5. rn is used to rename files.

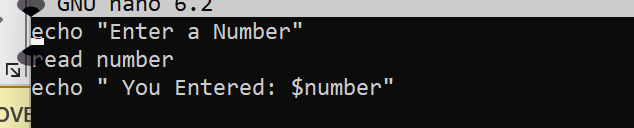
- “mv” command is used to rename files.

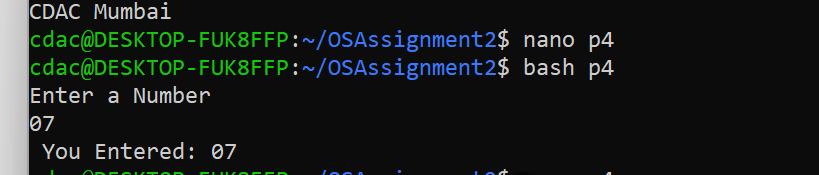
**Part C**

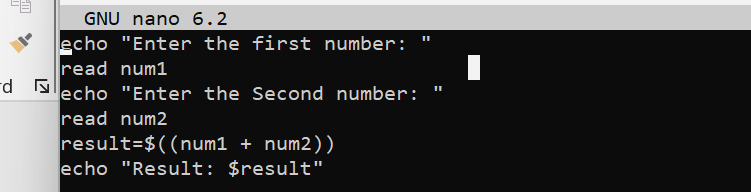
**Question 1:** Write a shell script that prints "Hello, World!" to the terminal.

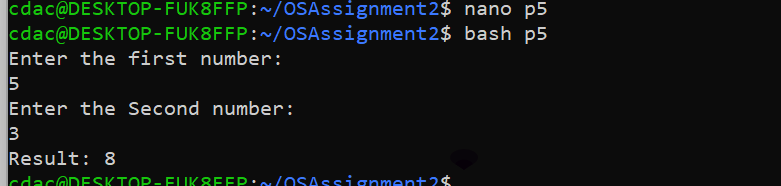


**Question 2:** Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

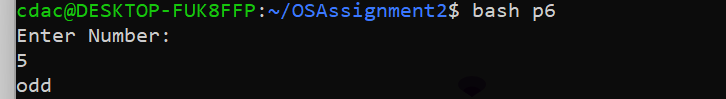
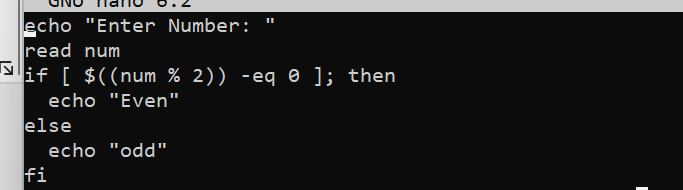
**Question 3:** Write a shell script that takes a number as input from the user and prints it.



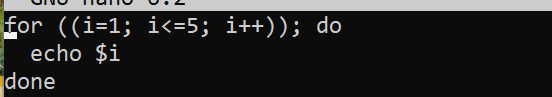
**Question 4:** Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.

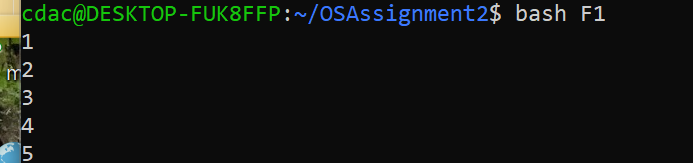


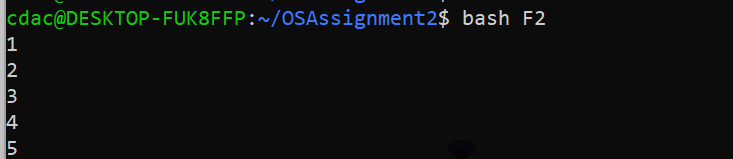
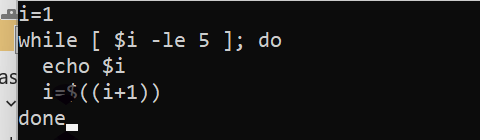
**Question 5:** Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

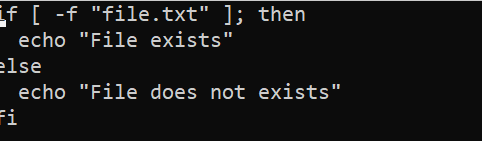


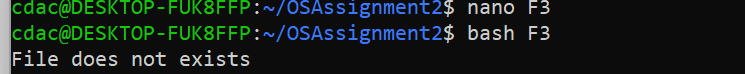
**Question 6:** Write a shell script that uses a for loop to print numbers from 1 to 5.



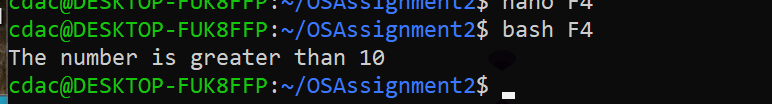
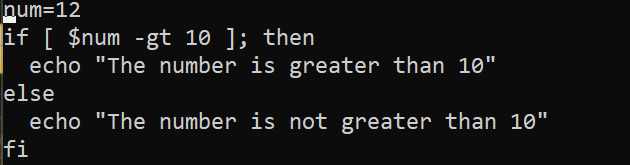


**Question 7:** Write a shell script that uses a while loop to print numbers from 1 to 5.

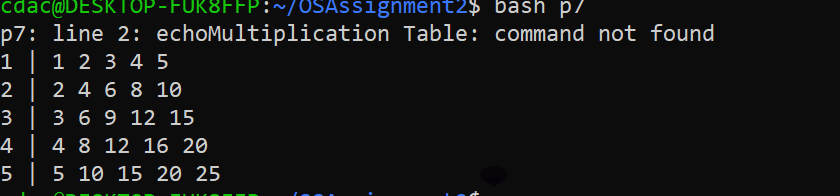
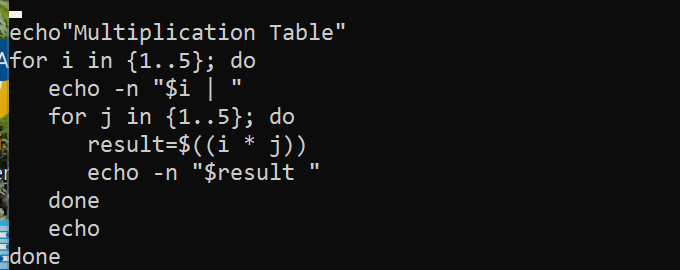
**Question 8:** Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".



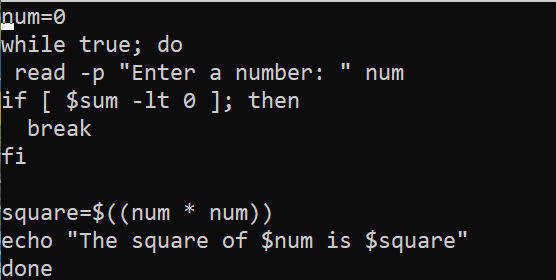
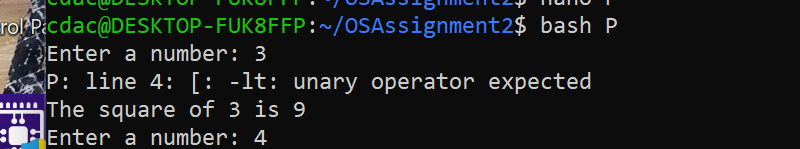
**Question 9:** Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.



**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:

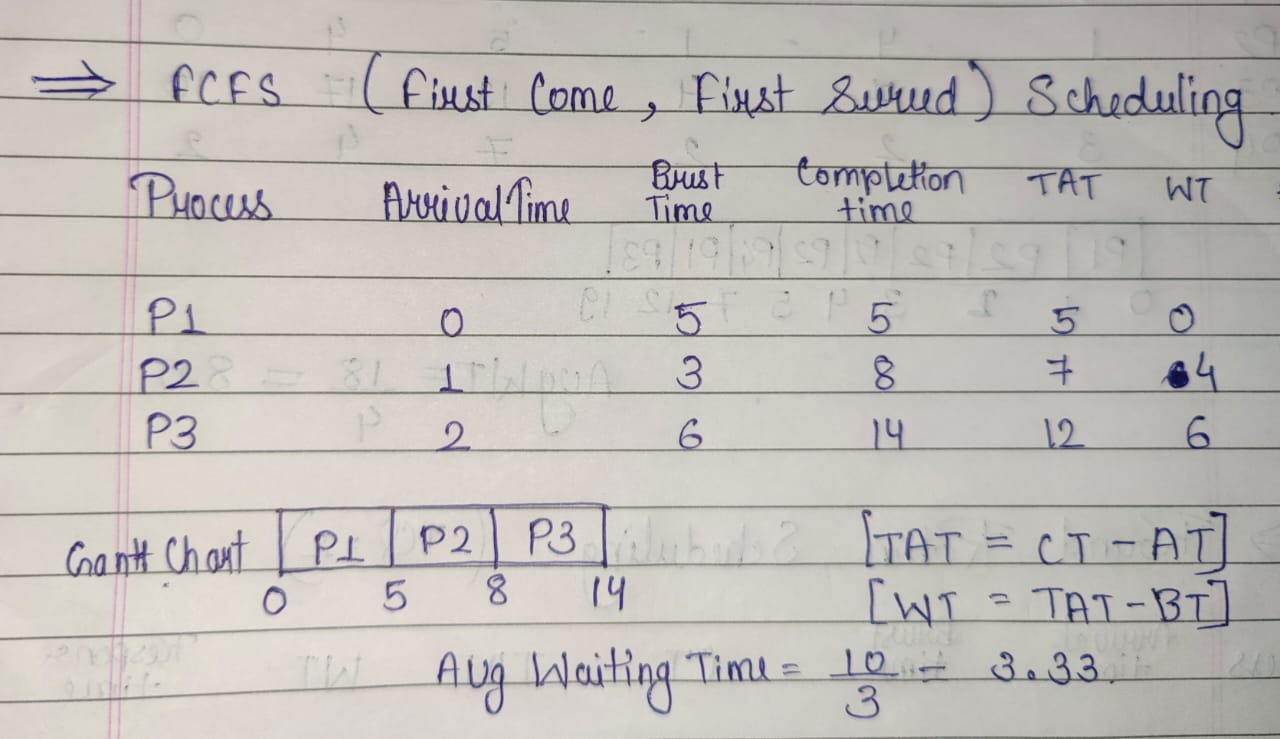
| Process | Arrival Time | Burst Time |

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

| 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 |

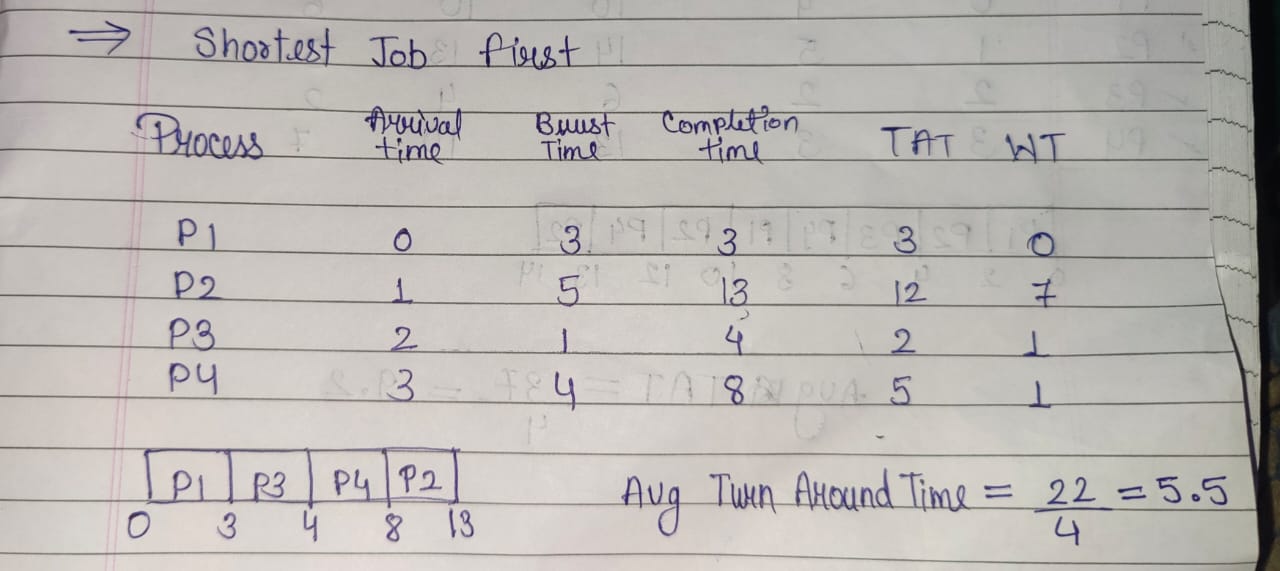
|---------|--------------|------------|

| P1 | 0 | 3 |

| P2 | 1 | 5 |

| P3 | 2 | 1 |

| P4 | 3 | 4 |

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

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

| Process | Arrival Time | Burst Time | Priority |

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

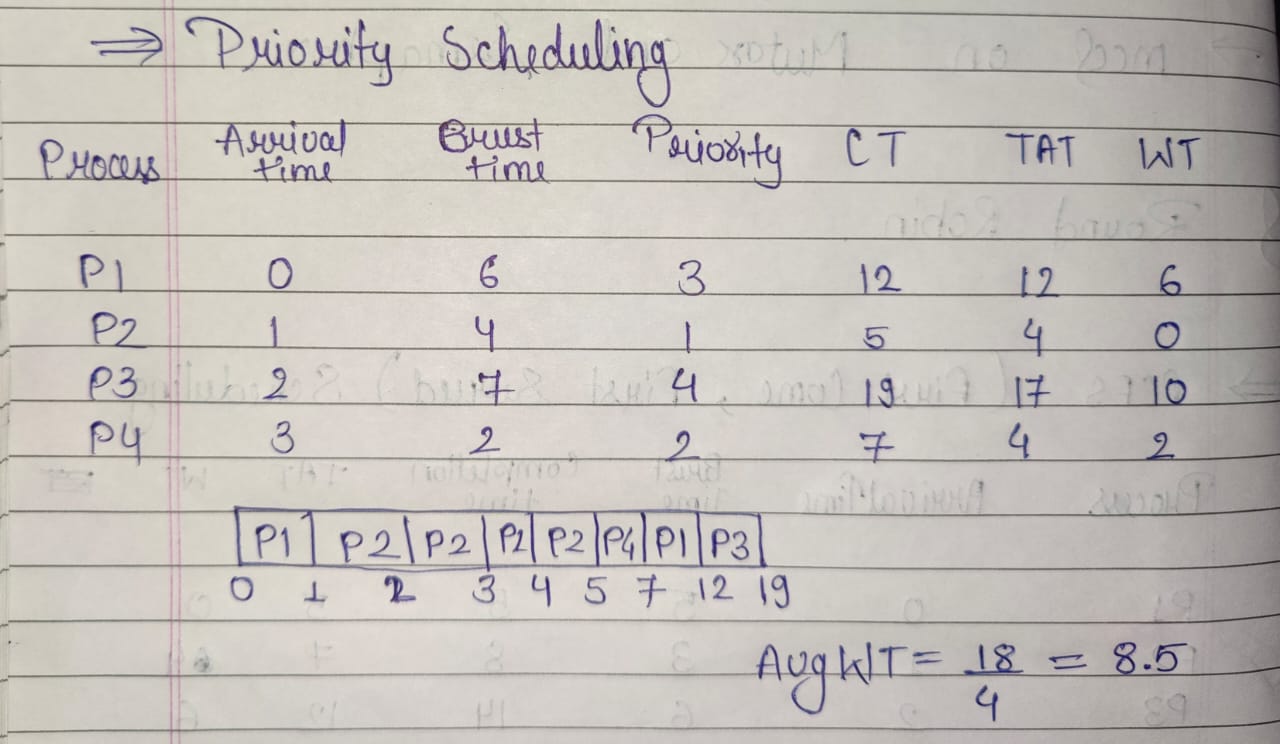
| P1 | 0 | 6 | 3 |

| 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.

