CDAC MUMBAI

Concepts of Operating System Assignment 2

**Name : Ankita Balasaheb Dhumal**

**Corse :DAC(Online)Feb25**

# Part A

What will the following commands do?

* echo "Hello, World!"

**Ans : This Command Prints the argument Hello world!**

* name="Productive"

**Ans : it stores value : Productive in variable : name**

* touch file.txt

**Ans : it will create new empty file : file.txt .**

* ls -a

**Ans :** **lists all files & directories and hidden files and directories from the current directory**

* rm file.txt

**Ans** : **It will remove file.txt from directory.**

* cp file1.txt file2.txt

**Ans:** **It will copy entire data of file1.txt into file2.txt if file2.txt is not present it will create it.**

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

**Ans : it will move file into given directory.**

* chmod 755 script.sh

**Ans** **: It changes the file permissions. Allows to owner to read, write, and execute (rwx=7).**

**allows the group and others to read and execute (r-x=5) 755.**

* grep "pattern" file.txt

**Ans** **:** **It will search for pattern word in file.txt.**

* kill PID

**Ans : It will send termination signal to the process id.**

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

**Ans : this command did multiple task at a time.first its create mkdir directory,then direct into it, then created empty file name as file.txt then print hello world into it and displayed contain of file.txt.**

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

**Ans : It will combines two files, removes duplicate data also arrange it alphabetically.**

* ls -l | grep "^d"

**Ans : ls -l will displays list of files and directories with permission information, it will filter the output and grep d find and shows lines starts with d.**

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

**Ans : it will recursively search for pattern from directory.**

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

**Ans : display combine both files in alphabetic order and shows only data which prasends in both files.**

* chmod 644 file.txt

**Ans : it sets permissions of file.txt as sets file permissions for file.txt so that : owner can read and write, the group can only read, others can only read.**

* cp -r source\_directory destination\_directory

**Ans : it copies one directory and its entire content into another directory.**

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

**Ans : it searched for all file that ends with .txt in directory.**

* chmod u+x file.txt

**Ans : it adds execute permission to user(owner) for file.txt.**

* echo $PATH

**Ans :Display current System’s environment variable.**

# Part B

Identify True or False:

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

**Ans : True.**

1. mv is used to move files and directories.

**Ans : True**.

1. cd is used to copy files and directories.

**Ans : False , cd is used to change the directory**

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

**Ans : True.**

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

**Ans : True.**

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

**Ans : True.**

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

**Ans : True.**

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

**Ans : True.**

Identify the Incorrect Commands:

1. chmodx is used to change file permissions.

**Ans : Incorrect. correct command is chmod to change file permission.**

1. cpy is used to copy files and directories.

**Ans : Incorrect. correct command is cp to copy files and directories**

1. mkfile is used to create a new file.

**Ans : Incorrect. there is no mkfile in Linux.**

1. catx is used to concatenate files.

**Ans : Incorrect. correct command is cat to concatenate file.**

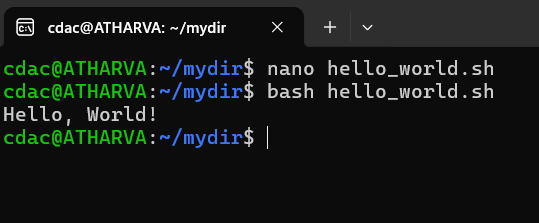
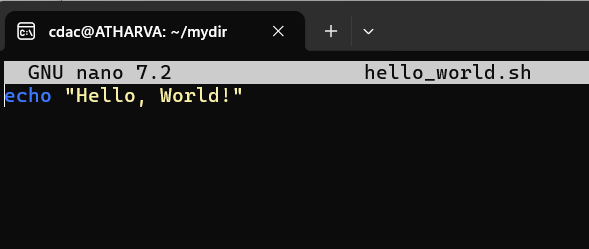
1. rn is used to rename

**Ans : Incorrect. correct command is mv to rename.**

# Part C

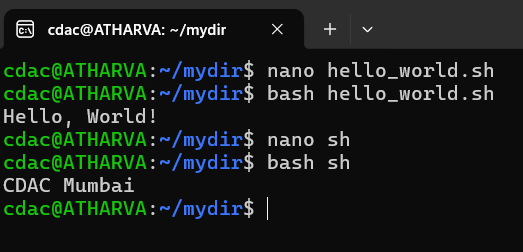
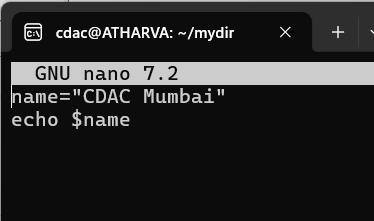
Question 1: Write a shell script that prints "Hello, World!" to the terminal.

**Ans :**

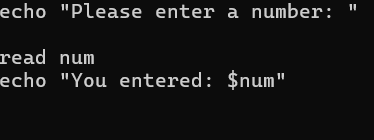
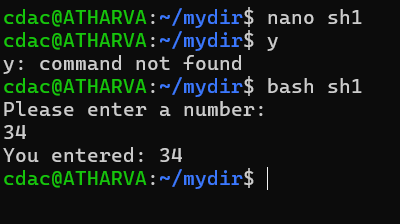
Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

**Ans :**

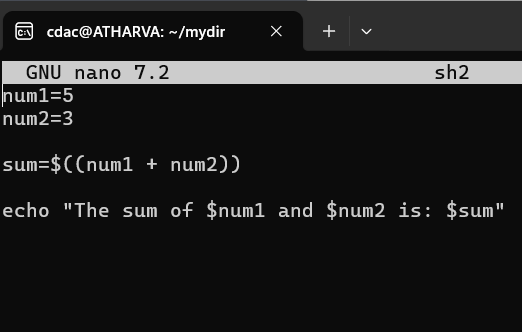
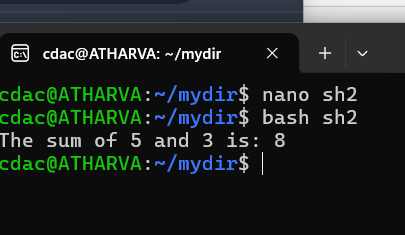
Question 3: Write a shell script that takes a number as input from the user and prints it.

**Ans :**

** **

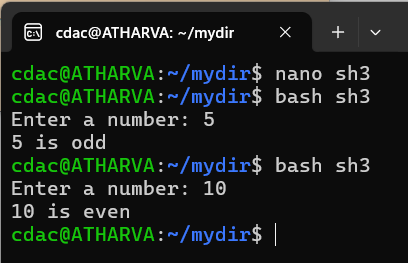
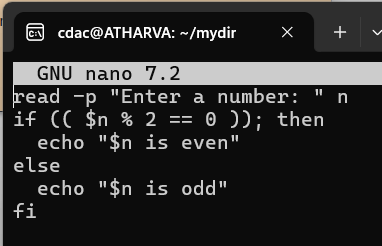
Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.

Ans :

****  ****

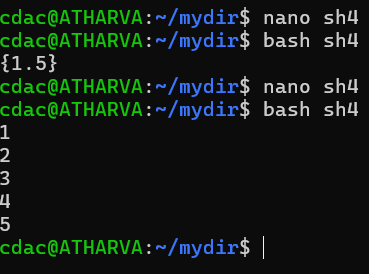
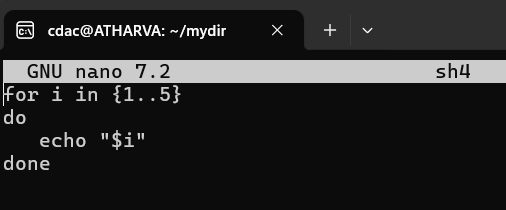
Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

**Ans :**

** **

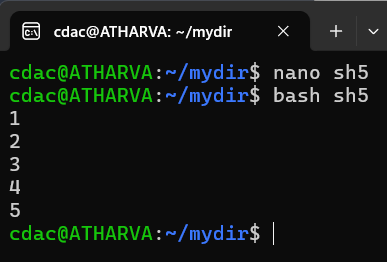
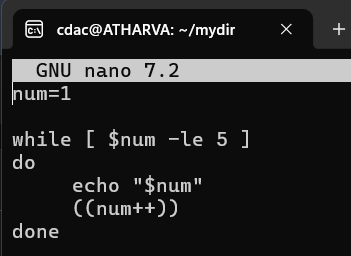
Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

**Ans :**

** **

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

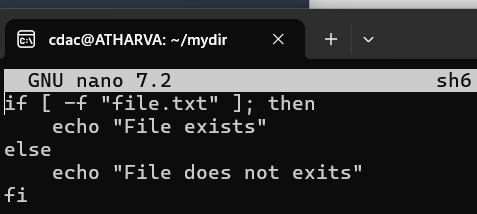
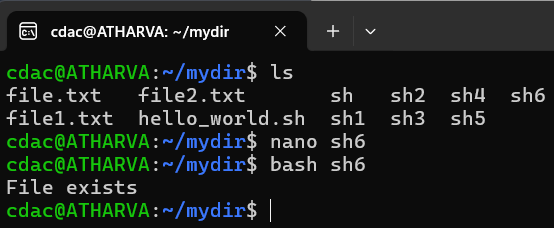
**Ans :**

** **

Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it

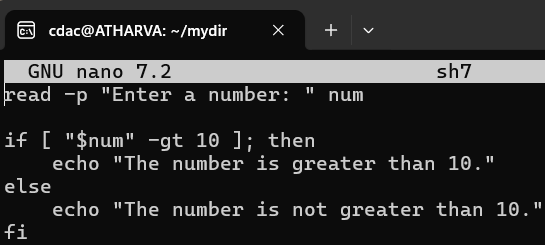
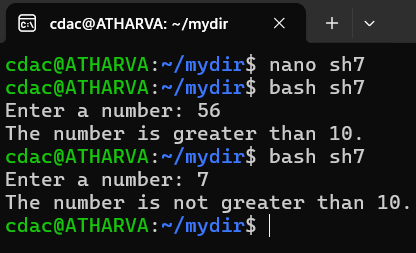
does, print "File exists", otherwise, print "File does not exist".

**Ans :**

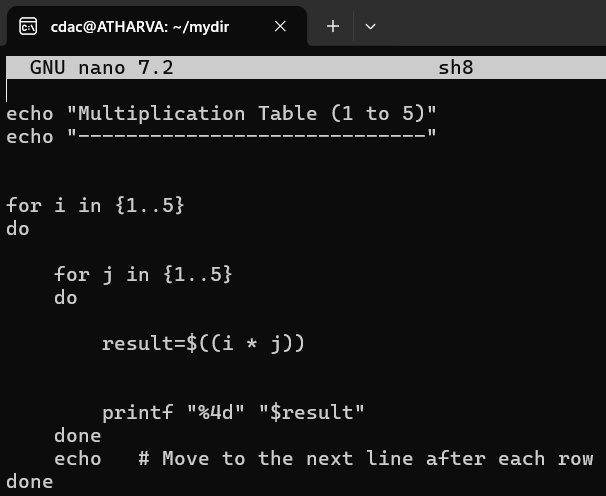
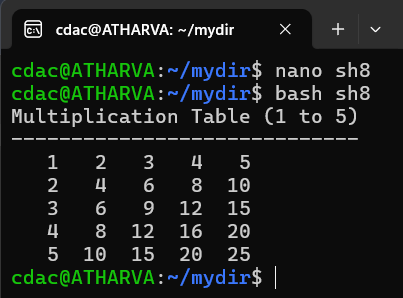
Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.

**Ans :**

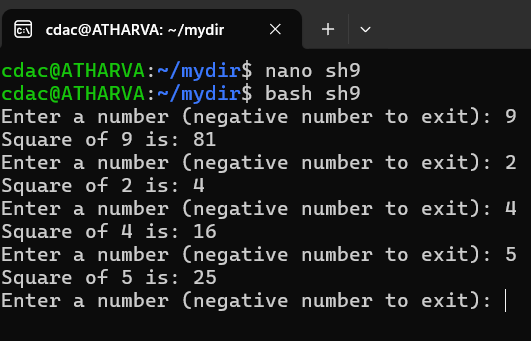
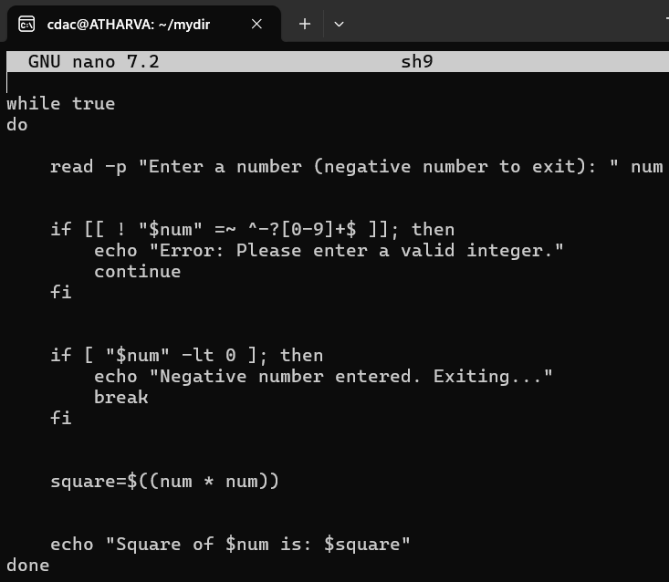
Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.

**Ans :**

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.

**Ans :**

# 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.

**Ans:**

| Process | Arrival Time | Burst Time | **Response Time | Turnaround Time |Wating Time**

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

| P1 | 0 | 5 **| 5 | 5 -0 = 5 | 0**

| P2 | 1 | 3 | **8 |8-1=7 | 4**

| P3 | 2 | 6 | **14 |14-2=12 | 6**

**Total Wating time=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 using Shortest Job First (SJF) scheduling.

**Ans:**

| Process | Arrival Time | Burst Time | **Wating Time| Turnaround Time | Wating Time**

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

| P1 | 0 | 3 | **3 | 3 | 0**

| P2 | 1 | 5 | **4 | 2 | 1**

| P3 | 2 | 1 | **8 | 5 | 1**

| P4 | 3 | 4 | **13 | 12 | 17**

**Average Turn Around Time = 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.

**Ans :**

| Process | Arrival Time | Burst Time | Priority | **Response Time | Trun Around Time| Wating Time**

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

| P1 | 0 | 6 | 3 | **0** | **6** | **0**

| P2 | 1 | 4 | 1 | **5** | **9** | **5**

| P3 | 2 | 7 | 4 | **7** | **9** | 7

| P4 | 3 | 2 | 2 | **10** | **17** | **10**

**Average Wating Time : 5.5**

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.

**Ans :**

| Process | Arrival Time | Burst Time| Response time| Wating Time|Turn Around time

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

| P1 | 0 | 4 | 0 | 6 | 10

| P2 | 1 | 5 | 1 | 8 | 13

| P3 | 2 | 2 | 2 | 2 | 4

| P4 | 3 | 3 | 3 | 7 | 10

**Average Turnaround Time : 9.25**

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?

**Ans :**

**Child Process X = 6**

**Parent Process X =6**