Part A

What will the following commands do?

echo "Hello, World!"

Explanation:

echo is a shell command used to print text to the terminal.

"Hello, World!" msg is gets printed printed.

What will the following commands do?

- name="Productive"

This sets a variable called name and the value "Productive" is assigned to variable name

- touch file.txt

Creates an empty file named file.txt

- ls -a

Lists all files and directories in the current directory, including hidden files (which start with . in Linux/Unix).Hidden files include .bashrc, .git, etc.

```
cdac@Pooja-Patil:~$ ls -a
.    .bash_history  .bashrc  .landscape  .local       .profile                    .viminfo  LinuxAssignment
..   .bash_logout   .cache   .lesshst     .motd_shown  .sudo_as_admin_successful  Feb25     filetxt
```

- rm file.txt

This attempts to remove (delete) a file named file.txt. This will permanently delete the file without confirmation

```
cdac@Pooja-Patil:~$ ls
Feb25  LinuxAssignment  file.txt  filetxt
cdac@Pooja-Patil:~$ rm file.txt
cdac@Pooja-Patil:~$ ls
Feb25  LinuxAssignment  filetxt
cdac@Pooja-Patil:~$ rm filetxt
cdac@Pooja-Patil:~$ ls
Feb25  LinuxAssignment
cdac@Pooja-Patil:~$
```

- cp file1.txt file2.txt

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

Moves file.txt to the specified directory. It can also be used to rename a file.

Rename : mv oldname.txt newname.txt

- chmod 755 script.sh

Changes permissions of script.sh to 755, it means read and exe permissions are given to everyone but onlu owner has write permission

- 7 (Owner) → Read, Write, Execute (rwx)
- 5 (Group) → Read, Execute (r-x)
- 5 (Others) → Read, Execute (r-x)

- grep "pattern" file.txt

Searches for "pattern" inside file.txt and prints matching lines.

- kill PID

Terminates a process with the specified Process ID **(**PID)

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

mkdir mydir → Creates a directory named mydir.

cd mydir → Moves into the mydir directory.

touch file.txt → Creates an empty file file.txt.

echo "Hello, World!" > file.txt → Writes "Hello, World!" into file.txt, overwriting any existing content.

cat file.txt → Displays the content of file.txt.

```
cdac@Pooja-Patil:~/LinuxAssignment$ mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat
file.txt
Hello, World!
cdac@Pooja-Patil:~/LinuxAssignment/mydir$
```

- ls -l | grep ".txt"

  ls -l → Lists files in long format (permissions, owner, size, date, etc.).

  grep ".txt" → Filters and displays only files with **".txt"** in their names.

```
cdac@Pooja-Patil:~/LinuxAssignment/mydir$ ls -l | grep ".txt"
-rw-r--r-- 1 cdac cdac 14 Mar  2 12:08 file.txt
cdac@Pooja-Patil:~/LinuxAssignment/mydir$
```

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

cat file1.txt file2.txt : Displays contents of both file1.txt and file2.txt.

sort : Sorts the combined content alphabetically.

uniq : Removes duplicate lines from the sorted output.

- ls -l | grep "^d"

ls -l : Lists all files and directories in long format.

grep "^d" : Filters only directories (lines starting with d, which indicates a directory).

```
cdac@Pooja-Patil:~/LinuxAssignment$  ls -l | grep "^d"
drwxr-xr-x 3 cdac  cdac 4096 Feb 27 13:48 docs
drwxr-xr-x 3 cdac  cdac 4096 Feb 27 14:34 lin
drwxr-xr-x 2 cdac  cdac 4096 Mar  2 12:08 mydir
drwxr-xr-x 3 cdac  cdac 4096 Feb 27 13:52 os1
cdac@Pooja-Patil:~/LinuxAssignment$
```

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

Recursively searches for "pattern" inside all files in /path/to/directory/.

This is useful for searching inside code, logs, or configuration files.

```
cdac@Pooja-Patil:~/LinuxAssignment$ grep -r "error" /etc
grep: /etc/credstore.encrypted: Permission denied
/etc/pam.d/common-password:# this avoids us returning an error just because nothing sets a success code
/etc/pam.d/common-account:# this avoids us returning an error just because nothing sets a success code
/etc/pam.d/common-session-noninteractive:# this avoids us returning an error just because nothing sets a success code
/etc/pam.d/common-auth:# this avoids us returning an error just because nothing sets a success code
/etc/pam.d/common-session:# this avoids us returning an error just because nothing sets a success code
grep: /etc/polkit-1/rules.d: Permission denied
grep: /etc/shadow-: Permission denied
grep: /etc/gshadow-: Permission denied
/etc/skel/.bashrc:# colored GCC warnings and errors
/etc/skel/.bashrc:#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'
/etc/skel/.bashrc:alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|t
 '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'
/etc/nanorc:## no colors by default, except for errorcolor and spotlightcolor.
/etc/nanorc:# set errorcolor bold,white,red
/etc/nanorc:# set errorcolor bold,white,red
grep: /etc/security/opasswd: Permission denied
/etc/X11/Xsession.d/50x11-common_determine-startup:  errormsg "$ERRMSG no session managers, no window managers, and no
/etc/X11/Xsession.d/20x11-common_process-args:              # fatal error
/etc/X11/Xsession.d/20x11-common_process-args:              errormsg "unable to launch failsafe X session ---" \
/etc/X11/Xsession.d/20x11-common_process-args:            # fatal error
/etc/X11/Xsession.d/20x11-common_process-args:            errormsg "unable to launch failsafe X session ---" \
/etc/X11/Xsession:errormsg () {
/etc/X11/Xsession:  # exit script with error
/etc/X11/Xsession:internal_errormsg () {
/etc/X11/Xsession:  # exit script with error; essentially a "THIS SHOULD NEVER HAPPEN" message
/etc/X11/Xsession:  # the user would have dismissed the error we want reported before seeing the
/etc/X11/Xsession:  errormsg "$*" \
/etc/X11/Xsession:          "package and the complete text of this error message to" \
/etc/X11/Xsession:ERRFILE=$HOME/.xsession-errors
/etc/X11/Xsession:# attempt to create an error file; abort if we cannot
/etc/X11/Xsession:             "\"$ERRFILE\"; look for session log/errors in" \
```

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

cat file1.txt file2.txt → Displays the contents of file1.txt and file2.txt.

sort → Sorts the combined content alphabetically.

uniq -d → Displays **only** duplicate lines appear in both files

---

- chmod 644 file.txt

Changes file permissions for file.txt to 644.

- **Owner (6)** : Read & Write
- **Group (4)** : Read-only
- **Others (4)** : Read-only

The file read and modified to owner.

The file can only be read by everyone else.

- cp -r source_directory destination_directory

cp -r : Copies a directory and its contents **recursively**.

source_directory → The folder you want to copy.

destination_directory → destination where the folder will get copied.

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

find /path/to/search : start to search inside the specified directory.

-name "*.txt" :Finds files that **end with .txt**.

- chmod u+x file.txt

chmod u+x : give permission of executing file to user or owner of file.txt

---

- echo $PATH

$PATH : Displays the system's PATH environment variable, which lists directories where the system looks for executable files.

```
cdac@Pooja-Patil:~/linux$ 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/Java/j
dk-11/bin:/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/Git/cmd:/mnt/c/TDM-GCC-32/bin:/
mnt/c/Program Files (x86)/Microsoft VS Code/bin:/mnt/c/Users/Pooja Patil/AppData/Local/Microsoft/WindowsApps:/snap/bin
```

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→ **Incorrect**

2. cpy is used to copy files and directories→ **Incorrect**

3. mkfile is used to create a new file →**Incorrect**

4. catx is used to concatenate files → **Incorrect**

5. rn is used to rename files →**Incorrect**

PART C

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

```
cdac@Pooja-Patil:~/Feb25$ mkdir UbuntuAssignment2 && cd UbuntuAssignment2
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ ls -l
total 0
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ nano sh1
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh1
Hello, World!
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$
```

Q2 Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

```
  GNU nano 7.2                                    sh2
name="CDAC Mumbai"

echo "name = $name"

                                   [ Read 3 lines ]
^G Help      ^C Write Out   ^W Where Is   ^K Cut       ^T Execute   ^C Location   M-U Undo     M-A Set Mark
```

```
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ nano sh2
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh2
name = CDAC Mumbai
```

Q3: Write a shell script that takes a number as input from the user and prints it.

```
  GNU nano 7.2
echo "enter the number"

read num

echo "the number is $num"
```

```
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ nano sh3
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh3
enter the number
67
the number is 67
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ nano sh3
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$
```

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

```
  GNU nano 7.2
echo "enter two numbers num1 and num2"

read num1
read num2

add=$((num1+num2))

echo "addition of $num1 and $num2 is=$add"
```

```
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh4
enter two numbers num1 and num2
3
5
addition of 3 and 5 is=8
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ nano sh4
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh4
```

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

```
echo "enter number"

read num1

if [ $(( num1 %2 )) -eq 0 ]

 then

  echo "$num1 is even"

else

  echo "$num1 is odd"

fi
```

```
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh5
enter number
6
6 is even
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh5
enter number
7
7 is odd
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ nano sh5
```

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

```
echo "the numbers from 1 to 5"

for((i=1;i<=5;i++))

 do

   echo " $i "

done
```

```
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh6
the numbers from 1 to 5
 1
 2
 3
 4
 5
```

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

```
i=1
echo "the numbers from 1 to 5"
while [ $i -le 5 ]
 do
  echo "$i"
 i=`expr $i + 1`
done
```

```
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ nano sh7
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh7
the numbers from 1 to 5
1
2
3
4
5
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ nano sh7
```

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".

```
if [ -f "file.txt" ]
 then
  echo "File.txt is exist"
 else
 echo "File.txt is not exist"
fi
```

```
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh8
File.txt is not exist
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ ls
bsh5  sh1  sh2  sh3  sh4  sh5  sh6  sh7  sh8
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$
```

Q9 .Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.

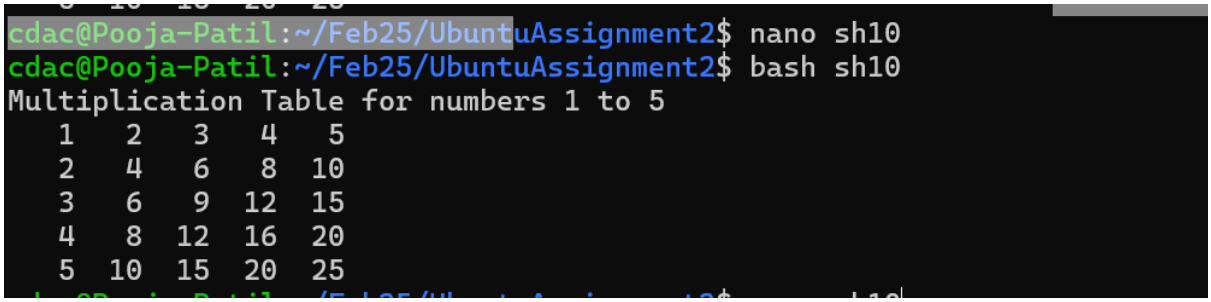
```
echo "Enter number"

read num

if [ $num -gt 10 ]

 then

 echo "number $num is greater than 10"

else

   echo "number $num is not greater than 10"

fi
```

```
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ nano sh9
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh9
Enter number
17
number 17 is greater than 10
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh9
Enter number
8
number 8 is not greater than 10
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$
```

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.

```
echo "Multiplication Table for numbers 1 to 5"

#echo "*************************"

for i in {1..5}

do

  for j in {1..5}

  do

    printf "%4d" $((i * j))

  done
```

```
  echo  #for new line
done
```



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.

```
while true;
 do
  # Read user input
  read -p "Enter a number: " num
  # Check if the number is negative
  if [ "$num" -lt 0 ]; then
    echo "Negative number entered, exit the loop"
    break
```

```
  fi
# Calculate and print the square of the number
  squ=$((num * num))
  echo "Square of $num is $squ"
done
```

```
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$ bash sh11
Enter a number: 7
Square of 7 is 49
Enter a number: 9
Square of 9 is 81
Enter a number: -1
Negative number entered, exit the loop
cdac@Pooja-Patil:~/Feb25/UbuntuAssignment2$
```

Part E

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?

the final values of x in the parent and child processes after the fork() call = 6
