

## Part A

### What will the following commands do?

- `echo "Hello, World!"`
  - I will print "Hello, World!" on console
- `name="Productive"`
  - It will assign string "Productive" to the variable "name"
- `touch file.txt`
  - It will create the .txt file
- `ls -a`
  - It is used to view all files and directories in the current directory and hidden files.
- `rm file.txt`
  - It is used to remove the file from current directory.
- `cp file1.txt file2.txt`
  - It is used to copy the content of file1 into file2.
- `mv file.txt /path/to/directory/`
  - Use for moving and renaming file and directory
- `chmod 755 script.sh`
  - It will give read, write and execute permission to owner(user) and gives read-execute to the group and other.
- `grep "pattern" file.txt`
  - This command in Linux is used to search for a specific text pattern in the file.
- `kill PID`
  - It is used to terminate a running process by sending a signal to it.
- `mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt`
  - Creates a new directory named mydir in the current working directory.
  - Changes the current working directory to the newly created mydir.
  - Creates an empty file named file.txt in the mydir directory
  - Writes the text "Hello, World!" into file.txt. The > operator redirects the output of echo to the file, overwriting its contents.
  - Displays the contents of file.txt in the terminal.
- `ls -l | grep ".txt"`
  - Lists all .txt files in the current directory in long format. It's a useful way to filter and display specific types of files.
- `cat file1.txt file2.txt | sort | uniq`
  - It will sort the content of both files and display the unique data only.
- `ls -l | grep "^d"`
  - It is used to list only directories in the current working directory.
- `grep -r "pattern" /path/to/directory/`
  - It is also used to search for a specific text pattern in the file.
- `cat file1.txt file2.txt | sort | uniq -d`
  - It is used to find duplicate lines across two files.
- `chmod 644 file.txt`
  - This is a common permission setting for files that need to be publicly readable but writable only by the owner.
- `cp -r source_directory destination_directory`
  - The command is used to recursively copy a directory and its contents to another location.
- `find /path/to/search -name "*.txt"`
  - It is used to find all .txt files in the given provided path.

- `chmod u+x file.txt`
  - The command adds execute permission for the owner of `file.txt`.
- `echo $PATH`
  - The command displays the current value of the `PATH` environment variable.

## **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.
  - False
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.
  - False

### **Identify the Incorrect Commands:**

1. `chmodx` is used to change file permissions.
  - There is no `chmodx` command in Linux.
  - The correct command to change file permissions is `chmod`.
2. `cpy` is used to copy files and directories.
  - There is no `cpy` command in Linux.
  - The correct command to copy files and directories is `cp`.

3. mkfile is used to create a new file.

- There is no mkfile command in standard Linux distributions.

- To create a new file, you can use commands like touch.

4. catx is used to concatenate files.

- The correct command to concatenate files is cat.

5. rn is used to rename files.

- The correct command to rename files is mv.

## Part C

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

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ touch file1.txt
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ nano file1.txt
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file.txt
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file1.txt
Hello, World!
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

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

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file6.txt
name="CDAC MUMBAI"
echo "The value of the variable is: $name"
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file6.txt
The value of the variable is: CDAC MUMBAI
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

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

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file1.txt
echo "Enter the number"
read num
echo $num
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file1.txt
Enter the number
2
2
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

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

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file7.txt
a=10
b=20
echo "The addition is: $((a+b))"
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file7.txt
The addition is: 30
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

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

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file5.txt
echo "Enter the number"
read Num
if [ $((Num % 2)) == 0 ]
then
    echo "$Num is an even number"
else
    echo "$Num is an odd number"
fi
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file5.txt
Enter the number
2
2 is an even number
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

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

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file8.txt
a=0
for a in 1 2 3 4 5
do
echo $a
done
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file8.txt
1
2
3
4
5
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

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

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file9.txt
a=1
while [ $a -lt 6 ]
do
echo $a
a=$(expr $a + 1)
done
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file9.txt
1
2
3
4
5
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

**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".

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file10.txt
file="file.txt"
if [ -e "$file" ]
then
    echo "File found"
else
    echo "File is not here"
fi
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file10.txt
File found
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

**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.

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file11.txt
a=2
if [ $a -gt 10 ]
then
    echo "The given number is greater than 10"
else
    echo "the given number is smaller than 10"
fi
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file11.txt
the given number is smaller than 10
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

**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.

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file12.txt
i=1
j=1
for i in 1 2 3 4 5
do
    for j in 1 2 3 4 5
    do
        echo -n "$i * $j = $((i * j))  "
    done
    echo
done
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file12.txt
1 * 1 = 1  1 * 2 = 2  1 * 3 = 3  1 * 4 = 4  1 * 5 = 5
2 * 1 = 2  2 * 2 = 4  2 * 3 = 6  2 * 4 = 8  2 * 5 = 10
3 * 1 = 3  3 * 2 = 6  3 * 3 = 9  3 * 4 = 12  3 * 5 = 15
4 * 1 = 4  4 * 2 = 8  4 * 3 = 12  4 * 4 = 16  4 * 5 = 20
5 * 1 = 5  5 * 2 = 10  5 * 3 = 15  5 * 4 = 20  5 * 5 = 25
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```

**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.

```
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ cat file13.txt
echo "Enter the number : "
read num
while [ $num -gt 0 ]
do
    square=$((num * num))
    echo "The square of $num is : $square"
    echo "Enter the number : "
    read num
done
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$ bash file13.txt
Enter the number : 
2
The square of 2 is : 4
Enter the number : 
3
The square of 3 is : 9
Enter the number : 
-1
sunbeam@sunbeam-HP-Notebook:~/MEHUL/LinuxAssignment/docs$
```