Assignment 2

Name: Suraj Balaram Patil (CDAC Kharghar)

Part A

What will the following commands do?

- echo "Hello, World!"

It Prints "Hello, World!" to the terminal.

```
cdac@DESKTOP-NOU1RRE:~$ echo "Hello World"
Hello World
```

- name="Productive"

Assigns the string "Productive" to the variable name.

```
cdac@DESKTOP-NOU1RRE:~$ name="Productive"
cdac@DESKTOP-NOU1RRE:~$
```

- touch file.txt

Creates an empty file named as file.txt

```
cdac@DESKTOP-NOU1RRE:~$ touch file.txt
cdac@DESKTOP-NOU1RRE:~$
```

- ls -a

Lists all files and directories, including hidden ones

```
cdac@DESKTOP-NOU1RRE:~$ ls -a
.   .bash_history  .bashrc     .lesshst  .motd_shown  .sudo_as_admin_successful  cdac
..  .bash_logout   .landscape  .local    .profile     OS_CDAC                    file.txt
cdac@DESKTOP-NOU1RRE:~$
```

- rm file.txt

Deletes file.txt permanently.

```
cdac@DESKTOP-NOU1RRE:~$ rm file.txt
cdac@DESKTOP-NOU1RRE:~$
```

- cp file1.txt file2.txt

Copies file1.txt to file2.txt

```
cdac@DESKTOP-NOU1RRE:~$ touch file1.txt
cdac@DESKTOP-NOU1RRE:~$ cp file1.txt file2.txt
cdac@DESKTOP-NOU1RRE:~$
```

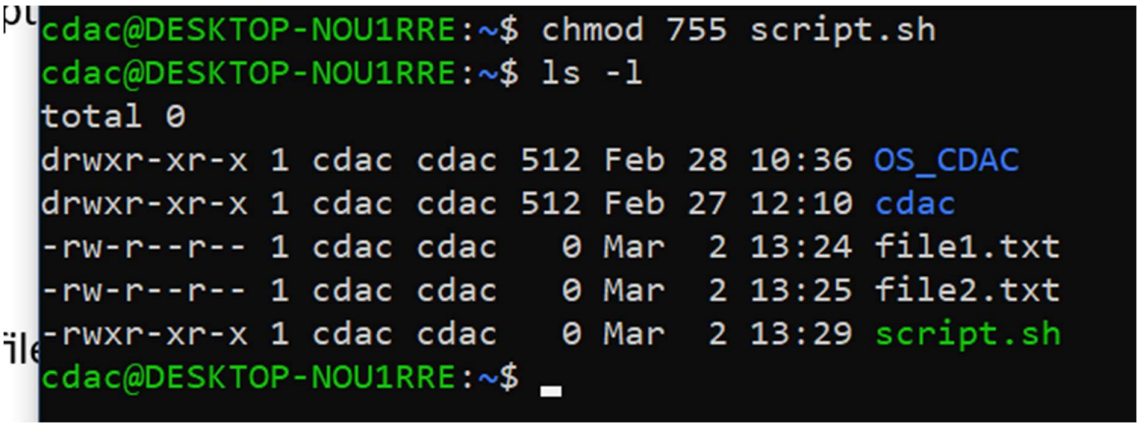
- mv file.txt /path/to/directory/

Moves file.txt to /path/to/directory/.

- chmod 755 script.sh

Changes the permissions of script.sh:

- Owner: read, write, execute (7).
- Group: read, execute (5).
- Others: read, execute (5).



- grep "pattern" file.txt

Searches for the string "pattern" in file.txt and prints matching lines.

- kill PID

Terminates the process with the specified Process ID (PID).

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

1. Creates a directory named mydir.
2. Change directory into mydir.
3. Creates an empty file file.txt.
4. Writes "Hello, World!" into file.txt.
5. Displays the contents of file.txt.

- ls -l | grep ".txt"

Lists all files in long format and filters those containing .txt.

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

Concatenates file1.txt and file2.txt, sorts the lines, and removes duplicates.

- ls -l | grep "^d"

Lists only directories those lines whose starting with d in ls -l.

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

search for "pattern" in all files under /path/to/directory/.

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

Concatenates both file1.txt and file2.txt, sorts the lines, and prints only duplicate lines.

- chmod 644 file.txt

Changes permissions of file.txt:

- Owner: read, write (6).
- Group: read (4).
- Others: read (4).

- cp -r source_directory destination_directory

Copy the source_directory to destination_directory

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

It searches for all .txt files under /path/to/search.

- chmod u+x file.txt

Grants execute permission to the file owner.

- echo $PATH

Displays the system's $PATH variable, which lists directories where executables are search.

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

Part B

Identify the Incorrect Commands:

1. chmodx is used to change file permissions.

- chmod command is used to change file permissions

2. cpy is used to copy files and directories.

- Cp command is used to copy files and directories

3. mkfile is used to create a new file.

- Touch command is used to create a new file. And mkdir used to create new directory

4. catx is used to concatenate files.

- cat command is used to concatenate files

5. rn is used to rename files.

- mv (move) command is used for renaming files.

Part C

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

```
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ touch hello.sh
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ ls
hello.sh
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ nano hello.sh
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ cat hello.sh
echo "Hello, World!"
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ bash hello.sh
Hello, World!
```

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

```
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ touch name.sh
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ nano name.sh
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ cat name.sh
Name="CDAC Mumbai"
echo $Name
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ bash name.sh
CDAC Mumbai
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$
```

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

```
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ cat input.sh
echo "Enter a Number : "
read a
echo "Number is "$a
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ bash input.sh
Enter a Number :
12
Number is 12
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$
```

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

```
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ cat add.sh
echo "Enter Num 1 : "
read num1
echo "Enter Num 2 : "
read num2
add=`expr $num1 + $num2`
echo "Addition of "$num1" and "$num2" is "$add
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ bash add.sh
Enter Num 1 :
34
Enter Num 2 :
66
Addition of 34 and 66 is 100
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$
```

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

```
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ nano even.sh
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ bash even.sh
Enter a Number :
23
23 is Odd
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$
```

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

```
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ nano for.sh
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ cat for.sh
for i in 1 2 3 4 5
do
        echo $i
done
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ bash for.sh
1
2
3
4
5
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$
```

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

```
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ cat while.sh
a=1
while [ $a -lt 6 ]
do
        echo $a
        a=`expr $a + 1`
done
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ bash while.sh
1
2
3
4
5
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$
```

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".

```
cdac@DESKTOP-NOU1RRE:~$ nano Q8.sh
cdac@DESKTOP-NOU1RRE:~$ cat Q8.sh
if [ -e file.txt ]
then
        echo "File Exists."
else
        echo "File doesn't Exists."
fi
cdac@DESKTOP-NOU1RRE:~$ bash Q8.sh
File doesn't Exists.
cdac@DESKTOP-NOU1RRE:~$
```

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.

```
cdac@DESKTOP-NOU1RRE:~$ nano greater.sh
cdac@DESKTOP-NOU1RRE:~$ cat greater.sh
echo "Enter a Number : "
read a
if [ $a -gt 10 ]
then
        echo "$a is greater than 10."
else
        if [ $a -eq 10 ]
        then
                echo "$a is Equal to 10."
        else
                echo "$a is Smaller than 10."
        fi
fi
cdac@DESKTOP-NOU1RRE:~$ bash greater.sh
Enter a Number :
12
12 is greater than 10.
cdac@DESKTOP-NOU1RRE:~$
```

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.

```
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ nano multiplication.sh
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ cat multiplication.sh
for i in {1..5}
do
        for j in {1..5}
        do
                ans=`expr $i \* $j`
                echo -n "$ans   "
        done
        echo
done
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ bash multiplication.sh
1   2   3   4   5
2   4   6   8   10
3   6   9   12   15
4   8   12   16   20
5   10   15   20   25
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$
```

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.

```
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ nano Loop.sh
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ cat Loop.sh
while [ true ]
do
        echo "Enter a Number : "
        read a
        if [ $a -lt 0 ]
        then
                break
        fi
done
echo "Execution Done.."
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$ bash Loop.sh
Enter a Number :
12
Enter a Number :
23
Enter a Number :
44
Enter a Number :
32
Enter a Number :
-1
Execution Done..
cdac@DESKTOP-NOU1RRE:~/OS_CDAC/Assignment_2$
```