

## Assignment 2

### Part A

What will the following commands do?

- echo "Hello, World!"
  - It will print Hello, World!
- name="Productive"
  - This command **name="Productive"** will create a string variable with name name and productive word is going to store in it.
  - When we do **Echo \$name**  
Output:-Productive
- touch file.txt
  - This command will create the file with name file.txt
- ls -a
  - List all files and directories in Alphabetical order.

Because of ' -a ' option entries starting with ' . .. ' are also printing at output i.e.all hidden files are also printed
- rm file.txt
  - Remove file.txt from given directory
- cp file1.txt file2.txt
  - Copy the content of file1.txt into file2.txt
- mv file.txt /path/to/directory/
  - move files and directories from one directory to another or to rename a file or directory.In this file.txt will move in specified directory.
- chmod 755 script.sh
  - This command will change the permission of user as all, for group it would be read and execute and for others also read and execute for file script.sh
- grep "pattern" file.txt
  - This command will search for lines matching a word " pattern "and print the matching lines to standard output.

Pattern on the screen is nice.

This is nice pattern.

I like this.

Output:- **Pattern** on the screen is nice.

This is nice **pattern**.

- kill PID

- This command will terminate a running process on a computer by specifying its Process ID (PID)

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

- && (logical AND) operator is used here which enables the user to run multiple commands in single command. **mkdir** will make directory with name **mydir** .&& operator will execute command 2 i.e. **cd mydir** if command 1 i.e. **mkdir mydir** executed successfully and then after **cd mydir** command **touch file.txt** will execute which will create file with name file.txt. after this "Hello, World!" will append in file.txt and then **cat file.txt** will print the contain of file.txt
- output :- Hello, World!

- ls -l | grep ".txt"

- **ls -l** command print the list of all files in directory but because of pipelining with **grep ".txt"** it will list files with .txt

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

- Because of pipelining with **sort** and **uniq** command. content of file2.txt and content of file1.txt are produces with sorted and unique manner .

- ls -l | grep "^d"

- **ls** command list the files and directories. **grep "^d"** command filters the output to show only lines that start with "d" which in the **ls -l** output indicates directories.

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

- Here **grep-r** command recursively search "pattern" in given directory. The lines which contain "pattern" will be printed.

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

- **cat file1.txt file2.txt** command will print content of file1 followed by content of file2.txt and **sort** command will sort the as it is pipe lined there.
- **Uniq -d** will only print duplicate lines, one for each group

- chmod 644 file.txt

- This command will change the mode of permission of user ,group and others.

- Here `chmod 644 file.txt` will allow user to read and write and for group and other it will be read permission for `file.txt`
  - `cp -r source_directory destination_directory`
    - This command will recursively copy the `source_directory` into `destination_directory`
  - `find /path/to/search -name "*.txt"`
    - This command is use to search the directories and file ending with `*.txt`
  - `chmod u+x file.txt`
    - this command will give execution permission of file `file.txt` to user.
  - `echo $PATH`
    - This command displays the value of system environment variable that stores directories where executable programs are located.
- 

## **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** , `cd` is use to change the 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.

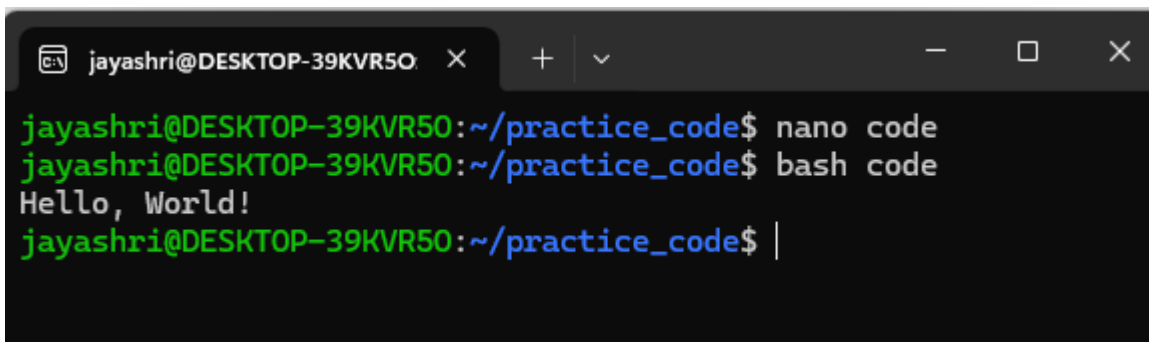
**:- False**

### Identify the Incorrect Commands:

1. **chmodx** is used to change file permissions.
  - **chmod** is used to change file permissions.
2. **cpy** is used to copy files and directories.
  - **cp** is used to copy files and directories.
3. **mkfile** is used to create a new file.
  - **touch** command is use to create a new file and **mkdir** is use to create new directory.
4. **catx** is used to concatenate files.
  - **cat** is used to concatenate files.
5. **rn** is used to rename files.
  - **mv** command is use to rename the file when file names are pass as arguments.

## Part c

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

A terminal window with a dark background. The title bar shows 'jayashri@DESKTOP-39KVR50'. The prompt is 'jayashri@DESKTOP-39KVR50:~/practice\_code\$'. The user enters 'nano code', then 'bash code', and the terminal outputs 'Hello, World!'. The prompt returns to 'jayashri@DESKTOP-39KVR50:~/practice\_code\$' with a cursor at the end.

```
jayashri@DESKTOP-39KVR50:~/practice_code$ nano code
jayashri@DESKTOP-39KVR50:~/practice_code$ bash code
Hello, World!
jayashri@DESKTOP-39KVR50:~/practice_code$ |
```

Code:-echo Hello,world!

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

```
jayashri@DESKTOP-39KVR50:~/practice_code$ nano code
jayashri@DESKTOP-39KVR50:~/practice_code$ bash code
Enter a name
CDAC Mumbai
CDAC Mumbai
jayashri@DESKTOP-39KVR50:~/practice_code$ |
```

Code :

```
echo "Enter a name"
```

```
read name
```

```
echo $name
```

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

```
jayashri@DESKTOP-39KVR50:~/practice_code$ nano code
jayashri@DESKTOP-39KVR50:~/practice_code$ bash code
Enter a number
15
15
jayashri@DESKTOP-39KVR50:~/practice_code$ |
```

Code:

```
echo "Enter a number"
```

```
read num
```

```
echo $num
```

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

```
jayashri@DESKTOP-39KVR50 X + v - □ X
jayashri@DESKTOP-39KVR50:~/practice_code$ nano code
jayashri@DESKTOP-39KVR50:~/practice_code$ bash code
Enter a number
5
Enter a number
3
sum is 8
jayashri@DESKTOP-39KVR50:~/practice_code$ |
```

Code:

```
echo "Enter a number"
read num1
echo "Enter a number"
read num2
result=`expr $num1 + $num2`
```

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

Code:

```
echo "Enter a number"
read $num
If (( $num % 2 == 0 ))
then
echo Even Number
else
echo Odd Number
fi
```

```
jayashri@DESKTOP-39KVR50 X + v - □ X
jayashri@DESKTOP-39KVR50:~/practice_code$ nano code
jayashri@DESKTOP-39KVR50:~/practice_code$ bash
jayashri@DESKTOP-39KVR50:~/practice_code$ bash code
Enter a number
15
Odd number
jayashri@DESKTOP-39KVR50:~/practice_code$ |
```

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

```
jayashri@DESKTOP-39KVR50: ~/practice_code$ nano code
jayashri@DESKTOP-39KVR50: ~/practice_code$ bash code

1
2
3
4
5
jayashri@DESKTOP-39KVR50: ~/practice_code$ nano code
jayashri@DESKTOP-39KVR50: ~/practice_code$ |
```

Code:

```
for(( i=1 ; i<=5 ; i++))
do
echo $i
done
```

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

```
jayashri@DESKTOP-39KVR50: ~/practice_code$ nano code
jayashri@DESKTOP-39KVR50: ~/practice_code$ bash code

1
2
3
4
5
jayashri@DESKTOP-39KVR50: ~/practice_code$ |
```

Code:

```
i=1
while [ $i -le 5 ]
do
echo $i
i=`expr $i + 1 `
done
```

**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".

```
jayashri@DESKTOP-39KVR50: ~/practice_code$ nano code
jayashri@DESKTOP-39KVR50: ~/practice_code$ cat code
if [ -e file.txt ]
then
echo "File exists"
else
echo "File doesn't exist"
fi
jayashri@DESKTOP-39KVR50: ~/practice_code$ bash code
File exists
jayashri@DESKTOP-39KVR50: ~/practice_code$ |
```

**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.

```
jayashri@DESKTOP-39KVR50: ~/practice_code$ nano greater10
jayashri@DESKTOP-39KVR50: ~/practice_code$ cat greater10
echo "Enter a number "
read a
if [ $a -gt 10 ]
then
    echo "Number is greater than 10"
else
    if [ $a -eq 10 ]
    then
        echo "Number is equal to 10 "
    else
        echo "Number is lass than 10"
    fi
fi
jayashri@DESKTOP-39KVR50: ~/practice_code$ bash greater10
Enter a number
55
Number is greater than 10
jayashri@DESKTOP-39KVR50: ~/practice_code$ |
```

**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.

```
jayashri@DESKTOP-39KVR50: ~/practice_code$ cat tableupto5
for ((i=1 ; i <= 10 ; i++ ))
do
for((j=1 ; j<=5 ; j++))
do
result=`expr $i \* $j `
echo -n "$result"
done
echo
done
jayashri@DESKTOP-39KVR50:~/practice_code$ bash tableupto5
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
6      12     18     24     30
7      14     21     28     35
8      16     24     32     40
9      18     27     36     45
10     20     30     40     50
jayashri@DESKTOP-39KVR50:~/practice_code$ |
```