Concepts of Operating System
Assignment 2
Part A

What will the following commands do?

1. echo "Hello, World!"
   Prints Hello, World!
2. name="Productive"
   Assigns variable
3. touch file.txt
   Creates file
4. ls -a
   List all files in cwd
5. rm file.txt
   Delete file
6. cp file1.txt file2.txt
   Copy & Override file1 to file2
7. mv file.txt /path/to/directory/
   Move file
8. chmod 755 script.sh
   Give permission of file to user=rwx, group=r-x, other=r-x
9. grep "pattern" file.txt
   Search word in file
10. kill PID
    Terminate process with given ID
11. mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
    Create directory mydir
    Go in mydir
    Create file
    Add message into file
    Show content in file
12. ls -l | grep ".txt"
    List file with details only with extension .txt
13. cat file1.txt file2.txt | sort | uniq
    Show unique records with sorting
14. ls -l | grep "^d"
    Only show detailed list of directories
15. grep -r "pattern" /path/to/directory/

Recursively search word

16. cat file1.txt file2.txt | sort | uniq –d
    Concatenate Sort and only show duplicate lines
17. chmod 644 file.txt
    Changes permissions
18. cp -r source_directory destination_directory
    Recursively copy
19. find /path/to/search -name "*.txt"
    Search files only with .txt name
20. chmod u+x file.txt
    Give execute permission to user
21. echo $PATH
    Show Path. It is 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. cd- change directory. cp-copy
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.
   chmod

2. cpy is used to copy files and directories.
   cp
3. mkfile is used to create a new file.
   touch
4. catx is used to concatenate files.
   cat
5. rn is used to rename files.
   mv

Part C

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

➔ echo "Hello, World!"

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

➔ name="CDAC Mumbai"
➔ echo "$name"

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

➔ echo "$name"

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

➔ read num
➔ echo "$num"

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

➔ var1=5
➔ var2=3
➔ sum=$((var1 + var2))
➔ echo "$sum"

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

➔ for i in 1 2 3 4 5
➔ do
➔ echo "$i"
➔ done

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

```
➔ num=1
➔
➔ while [ $num -le 5 ]
➔ do
➔     echo "$num"
➔     num=$((num + 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".

```
➔ if [ -f "file.txt" ]; then
➔     echo "File exists"
➔ else
➔     echo "File does not exist"
➔ fi
```

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.

```
➔ read num
➔
➔ if [ $num -gt 10 ]; then
➔     echo "Greater than 10 or 10"
➔ else
➔     echo "Smaller than 10"
➔ Fi
```

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.

```
➔ num=0
➔
➔ while [ 1 ]
➔ do
➔         read num
➔         if [ $num -lt 0 ]; then
➔                 echo "Negative Number"
```

➔ break
➔ else
➔ echo "Square: $((num * num))"
➔ fi
➔ Done