Concepts of Operating System
Assignment 2

Part A

What will the following commands do?

1) echo "Hello, World!"

Ans:- This will print Hello, World! On the Terminal

2) name="Productive"

Ans:- This will assign String value(Productive) in variable name

3) touch file.txt

Ans:- This will create a new file in current directory

4) ls-a

Ans:-This command will list all files and directories including Hidden files.

5) rm file.txt

Ans:-This will delete the file.txt from the current directory.

6) cp file1.txt file2.txt

Ans:- copy file1.txt file content into the file2.txt

7) mv file.txt /path/to/directory/

Ans:-This will move file.txt to the desired path.

8) chmod 755 script.sh

Ans:-This will allow rwx permission to owner, rx permission to group, rx permission to other to access the script.sh file.

9) grep "pattern" file.txt

Ans:-This will search for the pattern in file.txt file

10) kill PID

Ans:-This will terminate the process with given process ID

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

Ans:-mkdir will create mydir directory
cd mydir will get inside mydir
touch file.txt will create file in mydir directory
hello world will get inside (>)in file.txt
cat will display the file content of file.txt file

12) ls -l | grep ".txt"

Ans:-ls -l will list all files with details
| will push output of ls -l to grep
grep will filter line with .txt

13) cat file1.txt file2.txt | sort | uniq

Ans:-cat will display file1 file2 content by sorting and make the unique value if there are duplicates

14) ls -l | grep "^d"

Ans:- ls -l will list all the files with properties

grep "^d" will filter the lines starting with d

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

Ans:-Recursive search for "pattern" in all subdirectories

16) cat file1.txt file2.txt | sort | uniq –d

Ans:- Cat will concatenate both the file sort the file and display only duplicate line in the files

17) chmod 644 file.txt

Ans:-This will change permission of file.txt to u-rw , g-r , o-r.

18) cp -r source_directory destination_directory

Ans:-This will copy the file recursively from the source directory to destination directory.

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

Ans:-This will find the files ending with .txt

20) chmod u+x file.txt

Ans:-This will change the permission of the user to execute the file.txt

21) echo $PATH

Ans:-This will list path of the executable directories.

Part B

Identify True or False:

1)ls is used to list files and directories in a directory.

Ans:-True

2)mv is used to move files and directories.

Ans:-True

3)cd is used to copy files and directories.

Ans:-False

4)pwd stands for "print working directory" and displays the current directory.

Ans:-False

5)grep is used to search for patterns in files.

Ans:-True

6)chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.

Ans:-True

7)mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist.

Ans:-True

8)rm -rf file.txt deletes a file forcefully without confirmation.

Ans:-True

Identify the Incorrect Commands:

1)chmodx is used to change file permissions.
Ans:- chmod is correct command.

2)cpy is used to copy files and directories.

Ans:-cp is the correct command to copy file and directories

3)mkfile is used to create a new file.

Ans:- misfile will create the file of specific size.

4)catx is used to concatenate files.

Ans:- cat is used to concatenate files.

5)rn is used to rename files.

Ans:-mv is used to Rename the file.

Part C

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

Ans:-

```
duplicate.txt   file1           number.txt      threeNumSum
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment % nano hello
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment % bash hello
Hello, World!
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment %
```

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

```
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment % nano variable
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment % bash variable
CDAC Mumbai
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment % cat variable
name="CDAC Mumbai"
echo $name
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment % bash variable
CDAC Mumbai
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment %
```

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

```
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment % nano input
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment % cat input
echo Enter the Number
read num

echo "The number you entered is $num"
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment % bash input
Enter the Number
6
The number you entered is 6
saibabapc@MAHESHWARs-MacBook-Air LinuxAssignment %
```

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

```
saibabapc@MAHESHWARs-MacBook-Air coding % cat sum2Num
echo Enter the Number
read num1
echo Enter the Number
read num2

result=$((num1+num2))

echo "The sum of your numbers is $result"
saibabapc@MAHESHWARs-MacBook-Air coding % bash sum2Num
Enter the Number
5
Enter the Number
3
The sum of your numbers is 8
saibabapc@MAHESHWARs-MacBook-Air coding %
```

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

```
[saibabapc@MAHESHWARs-MacBook-Air coding % nano EvenOdd
[saibabapc@MAHESHWARs-MacBook-Air coding % cat EvenOdd
echo Enter the Number
read num

if ((num % 2 ==0 ));then
        echo "Even"
else
        echo "Odd"
fi
[saibabapc@MAHESHWARs-MacBook-Air coding % bash EvenOdd
Enter the Number
5
Odd
saibabapc@MAHESHWARs-MacBook-Air coding %
```

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

```
[saibabapc@MAHESHWARs-MacBook-Air coding % nano print1To5
[saibabapc@MAHESHWARs-MacBook-Air coding % cat print1To5
a=0

for a in 1 2 3 4 5
do

        echo "$a"
done
[saibabapc@MAHESHWARs-MacBook-Air coding % bash print1To5
1
2
3
4
5
saibabapc@MAHESHWARs-MacBook-Air coding %
```

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

```
[saibabapc@MAHESHWARs-MacBook-Air coding % nano while1To5
[saibabapc@MAHESHWARs-MacBook-Air coding % cat while1To5
a=0

while ((a != 6))
do
        echo "$a"
        ((a++))
done

[saibabapc@MAHESHWARs-MacBook-Air coding % bash while1To5
0
1
2
3
4
5
saibabapc@MAHESHWARs-MacBook-Air coding %
```

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".

```
saibabapc@MAHESHWARs-MacBook-Air coding % nano fileCheck
saibabapc@MAHESHWARs-MacBook-Air coding % cat fileCheck
if [ -f file.txt ];then
        echo "File.txt exist"
else
        ECHO "File.txt does not exist"
fi
saibabapc@MAHESHWARs-MacBook-Air coding % bash fileCheck
File.txt does not exist
saibabapc@MAHESHWARs-MacBook-Air coding %
```

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.

```
saibabapc@MAHESHWARs-MacBook-Air coding % nano checkNGrt
saibabapc@MAHESHWARs-MacBook-Air coding % cat checkNGrt
echo Enter the number
read num

if [ $num -gt 10 ];then
        echo "$num is greater than 10"
else
        echo "$num is smaller than 10"
fi
saibabapc@MAHESHWARs-MacBook-Air coding % bash checkNGrt
Enter the number
11
11 is greater than 10
saibabapc@MAHESHWARs-MacBook-Air coding %
```

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.

```
saibabapc@MAHESHWARs-MacBook-Air coding % nano multi
saibabapc@MAHESHWARs-MacBook-Air coding % cat multi
for (( i=1; i<=5; i++))
do
        for((j=1; j<=10; j++))
        do
                printf "%4d" "$((i * j))"
        done
        echo
done
saibabapc@MAHESHWARs-MacBook-Air coding % bash multi
   1   2   3   4   5   6   7   8   9  10
   2   4   6   8  10  12  14  16  18  20
   3   6   9  12  15  18  21  24  27  30
   4   8  12  16  20  24  28  32  36  40
   5  10  15  20  25  30  35  40  45  50
saibabapc@MAHESHWARs-MacBook-Air coding %
```

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.

```
saibabapc@MAHESHWARs-MacBook-Air coding % nano InpNum
saibabapc@MAHESHWARs-MacBook-Air coding % cat InpNum
while true
do
        echo "Enter the number"
        read num

        if [ $num -lt 0 ];then
                echo "You number is negative ,you are out"
                break
        fi

        square="$((num*num))"
        echo "Square of your num is $square"
done
saibabapc@MAHESHWARs-MacBook-Air coding % bash InpNum
Enter the number
4
Square of your num is 16
Enter the number
-2
You number is negative ,you are out
saibabapc@MAHESHWARs-MacBook-Air coding %
```