

Part A

echo "Hello, World!"

prints Hello,World on terminal

name="Productive"

assign Productive to variable name

touch file.txt

creates file file.txt

ls -a

list files including hidden files and directory

rm file.txt

deletes file.txt

cp file1.txt file2.txt

copy file1.txt as file2.txt

mv file.txt /path/to/directory/

moves/cut file.txt into specified path

chmod 755 script.sh

gives all permission to user and read as well as execute permissions to group and others.

grep "pattern" file.txt

search pattern in file.txt

kill PID

kill process with id PID

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

creates mydir directory navigates to it then create file.txt and then redirect output of echo to file.txt and then print content of file.txt in terminal

ls -l | grep ".txt"

list all content and its output given as input to grep to search pattern .txt in it

cat file1.txt file2.txt | sort | uniq

display contents of both file then its redirected to sort to sort the text then removes duplicates and display it on terminal

ls -l | grep "^d"

list down all content and then filters the directory and displays it

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

recursively searches pattern in the directory

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

prints uniq statement.

chmod 644 file.txt

read and write permission to user and read permissions to group and user

cp -r source_directory destination_directory

recursively copies directory to destination

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

search files ending with .txt in specified directory

```
chmod u+x file.txt
```

gives execute permission to user for file.txt

```
echo $PATH
```

displays system path content

[illegible]

Part B

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**
-
1. chmodx is used to change file permissions. **Incorrect**
 2. cpy is used to copy files and directories. **Incorrect**
 3. mkfile is used to create a new file. **Incorrect**
 4. catx is used to concatenate files. **Incorrect**
 5. rn is used to rename files. **Incorrect**

Part C

Code

```
cdac@SOHELKHAN:~/mydir$ for file in `ls -tr *.sh`
> do
> echo -e "\ncat $file"
> cat $file
> done

cat hw.sh
echo Hello, World!

cat print_name.sh
#!/bin/bash
name=CDAC
echo $name

cat input_no.sh
#!/bin/bash
echo "Enter Number:"
read num
echo "you have entered: $num"

cat addion_no.sh
#!/bin/bash
echo "Enter first Number:"
read num1
echo -e "Enter second Number:"
read num2
res=`expr $num1 + $num2`
echo "Addition of two number is: $res"

cat even_odd.sh
#!/bin/bash
echo "Enter a number: "
read num
if [[ `expr $num % 2` -eq 0 ]]
then
echo "Even"
else
echo "Odd"
fi

cat for_loop.sh
#!/bin/bash
for ((i=1; i<=5; i++))
do
    echo $i
done

cat while_loop.sh
#!/bin/bash
i=1
while [ $i -le 5 ]
do
    echo $i
    i=`expr $i + 1`
done

cat file_exists.sh
#!/bin/bash
if [[ -e "file.txt" ]]
then
    echo "File exists"
else
    echo "File does not exist"
fi
```

```
cat no_greater_10.sh
#!/bin/bash
echo "Enter a number:"
read num
if [[ $num -gt 10 ]]
then
echo "The number is greater than 10."
else
echo "The number is not greater than 10."
fi
```

```
cat nested_for.sh
#!/bin/bash
for ((i=1; i<=5; i++))
do
for ((j=1; j<=5; j++))
do
echo -n "`expr $i \* $j`  "
done
echo
done
```

```
cat while_inf.sh
#!/bin/bash
while true
do
echo "Enter a number:"
read number
if [[ $number -lt 0 ]]
then
break
fi
square=`expr $number \* $number`
echo "The square of $number is $square"
done
sdac@SOHEIL KHAN:~/mydir$
```

Output

```
cdac@SOHELKHAN:~/mydir$ ./hw.sh
./hw.sh: line 1: Hello,: command not found
cdac@SOHELKHAN:~/mydir$ echo echo "Hello, World!" > hw.sh
cdac@SOHELKHAN:~/mydir$ ./hw.sh
Hello, World!
cdac@SOHELKHAN:~/mydir$
cdac@SOHELKHAN:~/mydir$ ./hw.sh
Hello, World!
cdac@SOHELKHAN:~/mydir$ echo echo "Hello, World!" > hw.sh
cdac@SOHELKHAN:~/mydir$ chmod +x hw.sh
cdac@SOHELKHAN:~/mydir$ ./hw.sh
Hello, World!
cdac@SOHELKHAN:~/mydir$ nano print_name.sh
cdac@SOHELKHAN:~/mydir$ chmod +x print_name.sh
cdac@SOHELKHAN:~/mydir$ ./print_name.sh
CDAC
cdac@SOHELKHAN:~/mydir$ nano input_no.sh
cdac@SOHELKHAN:~/mydir$ cp print_name.sh input_no.sh
cdac@SOHELKHAN:~/mydir$ nano input_no.sh
cdac@SOHELKHAN:~/mydir$ chmod +x input_no.sh
cdac@SOHELKHAN:~/mydir$ ./input_no.sh
./input_no.sh: line 2: read: 'Enter Number,:': not a valid identifier
you have entered:
cdac@SOHELKHAN:~/mydir$ nano input_no.sh
cdac@SOHELKHAN:~/mydir$ ./input_no.sh
Enter Number:
10
you have entered: 10
cdac@SOHELKHAN:~/mydir$ cp input_no.sh addion_no.sh
cdac@SOHELKHAN:~/mydir$ nano addion_no.sh
cdac@SOHELKHAN:~/mydir$ chmod +x addion_no.sh
cdac@SOHELKHAN:~/mydir$ ./addion_no.sh
Enter first Number:
12
Enter second Number:
19
./addion_no.sh: line 6: unexpected EOF while looking for matching ``'
cdac@SOHELKHAN:~/mydir$ nano addion_no.sh
cdac@SOHELKHAN:~/mydir$ ./addion_no.sh
Enter first Number:
12
Enter second Number:
10
./addion_no.sh: line 6: res: command not found
Addition of two number is:
cdac@SOHELKHAN:~/mydir$ nano addion_no.sh
cdac@SOHELKHAN:~/mydir$ ./addion_no.sh
Enter first Number:
10
Enter second Number:
13
Addition of two number is: 26
cdac@SOHELKHAN:~/mydir$ nano addion_no.sh
cdac@SOHELKHAN:~/mydir$ nano addion_no.sh
cdac@SOHELKHAN:~/mydir$ ./addion_no.sh
Enter first Number:
10
Enter second Number:
12
Addition of two number is: 22
cdac@SOHELKHAN:~/mydir$ cp addion_no.sh even_odd.sh
cdac@SOHELKHAN:~/mydir$ nano even_odd.sh
cdac@SOHELKHAN:~/mydir$ chmod +x even_odd.sh
cdac@SOHELKHAN:~/mydir$ ./even_odd.sh
Enter a number:
23
./even_odd.sh: line 4: conditional binary operator expected
./even_odd.sh: line 4: syntax error near '%'
./even_odd.sh: line 4: `if [[ number % 2 == 0 ]]'
cdac@SOHELKHAN:~/mydir$ nano even_odd.sh
```

```

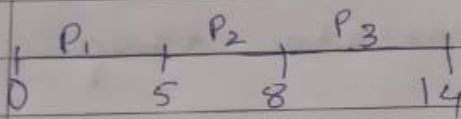
cdac@SOHELKHAN:~/mydir$ nano even_odd.sh
cdac@SOHELKHAN:~/mydir$ ./even_odd.sh
Enter a number:
23
./even_odd.sh: line 4: 23: command not found
Even
cdac@SOHELKHAN:~/mydir$ nano even_odd.sh
cdac@SOHELKHAN:~/mydir$ ./even_odd.sh
Enter a number:
23
Odd
cdac@SOHELKHAN:~/mydir$ ./even_odd.sh
Enter a number:
22
Even
cdac@SOHELKHAN:~/mydir$ nano for_loop.sh
cdac@SOHELKHAN:~/mydir$ chmod +x for_loop.sh
cdac@SOHELKHAN:~/mydir$ ./for_loop.sh
1
2
3
4
5
cdac@SOHELKHAN:~/mydir$ cp for_loop.sh while_loop.sh
cdac@SOHELKHAN:~/mydir$ nano while_loop.sh
cdac@SOHELKHAN:~/mydir$ chmod +x while_loop.sh
cdac@SOHELKHAN:~/mydir$ ./while_loop.sh
1
2
3
4
5
cdac@SOHELKHAN:~/mydir$ nano file_exists.sh
cdac@SOHELKHAN:~/mydir$ ./file.txt
./file.txt: line 1: Hello,: command not found
cdac@SOHELKHAN:~/mydir$ chmod +x file_exists.sh
cdac@SOHELKHAN:~/mydir$ ./file_exists.sh
File exists
cdac@SOHELKHAN:~/mydir$ nano no_greater_10.sh
cdac@SOHELKHAN:~/mydir$ chmod +x no_greater_10.sh
cdac@SOHELKHAN:~/mydir$ ./no_greater_10.sh
Enter a number:
9
The number is not greater than 10.
cdac@SOHELKHAN:~/mydir$ nano nested_for.sh
cdac@SOHELKHAN:~/mydir$ chmod +x nested_for.sh
cdac@SOHELKHAN:~/mydir$ ./nested_for.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@SOHELKHAN:~/mydir$ nano while_inf.sh
cdac@SOHELKHAN:~/mydir$ chmod +x while_
chmod: cannot access 'while_': No such file or directory
cdac@SOHELKHAN:~/mydir$ chmod +x while_inf.sh
cdac@SOHELKHAN:~/mydir$ ./while_inf.sh
Enter a number:
12
The square of 12 is 144
Enter a number:
23
The square of 23 is 529
Enter a number:
45
The square of 45 is 2025
Enter a number:
-10
cdac@SOHELKHAN:~/mydir$

```

1) Consider the following processes with arrival times & burst time

waiting time	Process	Arrival Time	Burst Time
0	P ₁	0	5
5	P ₂	0	3
8	P ₃	0	6

Calculate avg waiting time using FCFS scheduling.

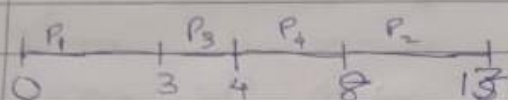


$$\text{Avg w.t.} = \frac{13}{3} = 4.33$$

2] Consider the following processes with arrival times & burst times: 1/1

Calculate the average Turnaround time using SJF scheduling.

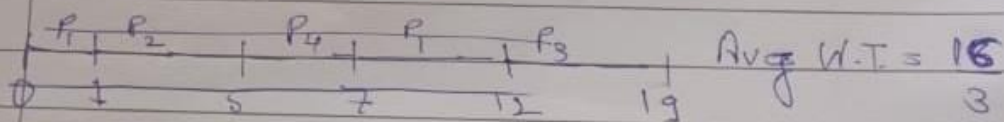
Process	Arr Time	Burst Time	Waiting Time	TAT
P ₁	0	3	0	3
P ₂	1	5	7	12
P ₃	2	1	1	2
P ₄	3	4	1	5



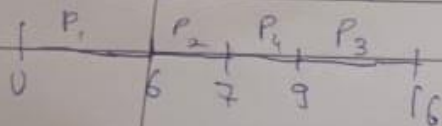
$$\text{Avg TAT} = \frac{22}{4} = 5.5$$

3] Consider following process with arrival time, burst times & priorities (lower number indicates higher priority). Average W.T. = ?

Process	Arrival Time	Burst Time	Priority	Waiting time
P ₁	0	5	3	0
P ₂	1	4	1	5
P ₃	2	7	4	4
P ₄	3	2	2	7



$$\text{Avg W.T.} = \frac{16}{3}$$



$$\text{Avg W.T.} = 5.33$$

4] Consider the following processes with arrival times and burst times, and the time quantum for Round Robin scheduling is 2

Process	Arrival time	Burst time	Waiting Time	T.A.T.
P ₁	0	4	6	10
P ₂	1	5	9	14
P ₃	2	2	2	4
P ₄	3	3	10	13

Calculate Turn around time using Round Robin
T.Q. = 2 unit

Timeline diagram showing process execution order: P₁, P₂, P₃, P₄, P₁, P₂, P₄, P₂, P₄. Time points are marked at 0, 2, 4, 6, 8, 10, 12, 13, 14, 15.

Avg. TAT = $\frac{38}{4} = 9.5$ $\frac{10+14+4+13}{4}$
 $= 10.25$

Consider a program that uses the fork() system call to create a child process. Initially, the parent process has a variable x with a value of 5. After forking, both the parent and child processes increment the value of x by 1. What will be the final values of x in the parent and child processes after the fork() call?

Final Value will 6.