

Part A

What will the following commands do?

- `echo "Hello, World!"`

It is used to print the output on terminal

```
cdac@ADITYA-AMBAR:~$ echo "Hello, World!"
Hello, World!
```

- `name="Productive"`

The name is a variable which contains the value as "Productive"

```
cdac@ADITYA-AMBAR:~$ name="Productive"
cdac@ADITYA-AMBAR:~$ echo $name
Productive
```

- `ls -a`

"ls" stands for list all files and `ls -a` means list all files and directories including hidden files

```
cdac@ADITYA-AMBAR:~$ ls -a
-i .bash_history .lessht .profile d2.txt duplicate.txt output2.txt
. .bash_logout .local .sudo_as_admin_successful d3.txt fruit.txt
.. .bashrc .motd_shown d1.txt d4.txt output.txt
```

- `touch file1.txt`: touch command is used to create a empty file
- `rm file1.txt`: rm command is used to delete a file

```
cdac@ADITYA-AMBAR:~$ ls
-i d1.txt d2.txt d3.txt d4.txt duplicate.txt fruit.txt output.txt output2.txt
cdac@ADITYA-AMBAR:~$ touch file1.txt
cdac@ADITYA-AMBAR:~$ ls
-i .bash_logout .lessht .motd_shown .sudo_as_admin_successful d2.txt d4.txt file1.txt output.txt
.bash_history .bashrc .local .profile d1.txt d3.txt duplicate.txt fruit.txt output2.txt
cdac@ADITYA-AMBAR:~$ ls
-i d1.txt d2.txt d3.txt d4.txt duplicate.txt file1.txt fruit.txt output.txt output2.txt
cdac@ADITYA-AMBAR:~$ rm file1.txt
cdac@ADITYA-AMBAR:~$ ls
-i d1.txt d2.txt d3.txt d4.txt duplicate.txt fruit.txt output.txt output2.txt
```

- `cp file1.txt file2.txt`

`cp` command is used to copy the contents file1 to file2 . Here file1 is source file1 and file2 is destination file.

```
cdac@ADITYA-AMBAR:~$ touch file1.txt
cdac@ADITYA-AMBAR:~$ nano file1.txt
cdac@ADITYA-AMBAR:~$ cat file1.txt
This is the content of file 1
cdac@ADITYA-AMBAR:~$ touch file2.txt
cdac@ADITYA-AMBAR:~$ cat file2.txt
cdac@ADITYA-AMBAR:~$ cp file1.txt file2.txt
cdac@ADITYA-AMBAR:~$ cat file2.txt
This is the content of file 1
```

- mv d1.txt /path/to/directory/

mv command is used to move the file from current location to other.

```
cdac@ADITYA-AMBAR:~$ pwd
/home/cdac
cdac@ADITYA-AMBAR:~$ ls
-i d1.txt d2.txt d3.txt d4.txt duplicate.txt file1.txt file2.txt fruit.txt output.txt output2.txt
cdac@ADITYA-AMBAR:~$ cat d1.txt
Hello everyone....
Hello Sir...
Hello mam...
cdac@ADITYA-AMBAR:~$ mv d1.txt /home/LinuxAssignment/
mv: cannot move 'd1.txt' to '/home/LinuxAssignment/d1.txt': Permission denied
cdac@ADITYA-AMBAR:~$ sudo mv d1.txt /home/LinuxAssignment/
[sudo] password for cdac:
cdac@ADITYA-AMBAR:~$ cd /home/LinuxAssignment/
cdac@ADITYA-AMBAR:/home/LinuxAssignment$ ls
d1.txt data.txt docs docs.zip file1.txt input.txt newdocs
```

- chmod 755 script.sh

chmod command is used to change the permissions. Before executing the command the Owner has read,write permission , Group and Other had read permission. After executing the command the Owner has all the permission, Group and Other has read and execute permission.

```
cdac@ADITYA-AMBAR:~$ nano script.sh
cdac@ADITYA-AMBAR:~$ bash script.sh
Aditya
My First Shell Program
cdac@ADITYA-AMBAR:~$ ls -l
total 0
-rw----- 1 cdac cdac 35 Aug 28 23:21 -i
-rw-r--r-- 1 cdac cdac  0 Aug 28 20:39 d2.txt
-rw-r--r-- 1 cdac cdac  0 Aug 28 20:39 d3.txt
-rw-r--r-- 1 cdac cdac  0 Aug 28 20:39 d4.txt
-rw-r--r-- 1 cdac cdac 43 Aug 28 23:18 duplicate.txt
-rw-r--r-- 1 cdac cdac 30 Aug 29 20:21 file1.txt
-rw-r--r-- 1 cdac cdac 30 Aug 29 20:22 file2.txt
-rw-r--r-- 1 cdac cdac 87 Aug 28 23:39 fruit.txt
-rw-r--r-- 1 cdac cdac 46 Aug 28 23:05 output.txt
-rw-r--r-- 1 cdac cdac 46 Aug 28 23:06 output2.txt
-rw-r--r-- 1 cdac cdac 40 Aug 29 20:33 script.sh
cdac@ADITYA-AMBAR:~$ chmod 755 script.sh
cdac@ADITYA-AMBAR:~$ ls -l
total 0
-rw----- 1 cdac cdac 35 Aug 28 23:21 -i
-rw-r--r-- 1 cdac cdac  0 Aug 28 20:39 d2.txt
-rw-r--r-- 1 cdac cdac  0 Aug 28 20:39 d3.txt
-rw-r--r-- 1 cdac cdac  0 Aug 28 20:39 d4.txt
-rw-r--r-- 1 cdac cdac 43 Aug 28 23:18 duplicate.txt
-rw-r--r-- 1 cdac cdac 30 Aug 29 20:21 file1.txt
-rw-r--r-- 1 cdac cdac 30 Aug 29 20:22 file2.txt
-rw-r--r-- 1 cdac cdac 87 Aug 28 23:39 fruit.txt
-rw-r--r-- 1 cdac cdac 46 Aug 28 23:05 output.txt
-rw-r--r-- 1 cdac cdac 46 Aug 28 23:06 output2.txt
-rwxr-xr-x 1 cdac cdac 40 Aug 29 20:33 script.sh
cdac@ADITYA-AMBAR:~$ |
```

- grep "pattern" file.txt

This command is used for pattern matching. The -w is used to find the strings in the pattern, and -l used to ignore the case.

```
cdac@ADITYA-AMBAR:~$ cat file1.txt
This is the content of file 1
cdac@ADITYA-AMBAR:~$ grep "is" file1.txt
This is the content of file 1
cdac@ADITYA-AMBAR:~$ grep "is" -w file1.txt
This is the content of file 1
```

- kill PID

ps command is used to check the process status and kill command is used to kill the process manually.

```
cdac@ADITYA-AMBAR:~$ ps
  PID TTY          TIME CMD
   62 tty1        00:00:00 bash
  197 tty1        00:00:00 ps
cdac@ADITYA-AMBAR:~$ kill 197
-bash: kill: (197) - No such process
cdac@ADITYA-AMBAR:~$ ps
  PID TTY          TIME CMD
   62 tty1        00:00:00 bash
  198 tty1        00:00:00 ps
```

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

“mkdir mydir” creates a new directory, “cd mydir” is used to change the directory to mydir, “touch file.txt” creates a empty file inside mydir, “echo” is used to display the output on command line, “>” redirection operator is used to put the result in “file.txt”, “cat” is used to display the contents and “&&” all the conditions are true then only the desired output will be displayed.

```
cdac@ADITYA-AMBAR:~$ mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
Hello, World!
cdac@ADITYA-AMBAR:~/mydir$ ls
file.txt
cdac@ADITYA-AMBAR:~/mydir$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 14 Aug 29 21:07 file.txt
```

- ls -l | grep ".txt"

ls command is used to list all the file and directory's in current directory, -l gives the detailed view including permission details, modified time etc

```
cdac@ADITYA-AMBAR:~/mydir$ ls -l | grep ".txt"
-rw-r--r-- 1 cdac cdac 14 Aug 29 21:07 file.txt
```

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

Display the unique contents form both the files.

```
cdac@ADITYA-AMBAR:~$ nano x1.txt
cdac@ADITYA-AMBAR:~$ nano x2.txt
cdac@ADITYA-AMBAR:~$ cat x1.txt
redbull
sting
nimbuzz
7up
coke
pepsi
cdac@ADITYA-AMBAR:~$ cat x2.txt
Elon
Mark
Steve
coke
nimbuzz
Redlabel
Kingfisher
cdac@ADITYA-AMBAR:~$ cat x1.txt x2.txt | sort | uniq
7up
Elon
Kingfisher
Mark
Redlabel
Steve
coke
nimbuzz
pepsi
redbull
sting
```

- `ls -l | grep "^d"`

```
cdac@ADITYA-AMBAR:~$ ls -l | grep "^d"
drwxr-xr-x 1 cdac cdac 512 Aug 29 21:07 mydir
cdac@ADITYA-AMBAR:~$ pwd
/home/cdac
```

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

this command recursively search the given pattern in the given path.

```
cdac@ADITYA-AMBAR:/home$ grep -r "a" /home/LinuxAssignment/
/home/LinuxAssignment/d1.txt:Hello mam...
/home/LinuxAssignment/input.txt:imagine you have a file named "input.txt" with text content.
/home/LinuxAssignment/input.txt:use a command to translate all lowercase letters to uppercase in "input.txt" and save the modified text in a new file named
"output.txt
cdac@ADITYA-AMBAR:/home$ grep -r -w "a" /home/LinuxAssignment/
/home/LinuxAssignment/input.txt:imagine you have a file named "input.txt" with text content.
/home/LinuxAssignment/input.txt:use a command to translate all lowercase letters to uppercase in "input.txt" and save the modified text in a new file named
"output.txt
```

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

Displays only duplicate word from both files.

```
cdac@ADITYA-AMBAR:~$ cat x1.txt x2.txt
redbull
sting
nimbuzz
7up
coke
pepsi
Elon
Mark
Steve
coke
nimbuzz
Redlabel
Kingfisher
cdac@ADITYA-AMBAR:~$ cat x1.txt x2.txt | sort | uniq -d
coke
nimbuzz
```

- `chmod 644 file.txt`

```
cdac@ADITYA-AMBAR:~$ ls -l script.sh
-rwxr-xr-x 1 cdac cdac 40 Aug 29 20:33 script.sh
cdac@ADITYA-AMBAR:~$ chmod 644 script.sh
cdac@ADITYA-AMBAR:~$ ls -l script.sh
-rw-r--r-- 1 cdac cdac 40 Aug 29 20:33 script.sh
```

- `cp -r source_directory destination_directory`

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment$ cp -r /home/LinuxAssignment/ /home/cdac/mydir/
cdac@ADITYA-AMBAR:/home/LinuxAssignment$ ls -l docs
total 0
-rwxr--r-- 1 user2 root 18 Aug 28 18:15 file2.txt
cdac@ADITYA-AMBAR:/home/LinuxAssignment$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 46 Aug 28 21:14 d1.txt
-rw-r--r-- 1 root root 199 Aug 28 22:24 data.txt
drwxr-xr-x 1 root root 512 Aug 28 18:21 docs
-rw-r--r-- 1 root root 160 Aug 28 21:45 docs.zip
-rw-r--r-- 1 root root 18 Aug 28 18:06 file1.txt
-rw-r--r-- 1 root root 199 Aug 28 22:45 input.txt
drwxr-xr-x 1 root root 512 Aug 28 21:47 newdocs
cdac@ADITYA-AMBAR:/home/LinuxAssignment$ cd mydir
-bash: cd: mydir: No such file or directory
cdac@ADITYA-AMBAR:/home/LinuxAssignment$ cd /home/cdac/mydir/
cdac@ADITYA-AMBAR:~/mydir$ ls =l
ls: cannot access '=l': No such file or directory
cdac@ADITYA-AMBAR:~/mydir$ ls -l
total 0
drwxr-xr-x 1 cdac cdac 512 Aug 29 22:07 LinuxAssignment
-rw-r--r-- 1 cdac cdac 14 Aug 29 21:07 file.txt
cdac@ADITYA-AMBAR:~/mydir$ |
```

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

```
cdac@ADITYA-AMBAR:/home$ find /home -name "*.txt"
/home/cdac/d2.txt
/home/cdac/d3.txt
/home/cdac/d4.txt
/home/cdac/duplicate.txt
/home/cdac/file1.txt
/home/cdac/file2.txt
/home/cdac/fruit.txt
/home/cdac/mydir/file.txt
/home/cdac/mydir/LinuxAssignment/d1.txt
/home/cdac/mydir/LinuxAssignment/data.txt
/home/cdac/mydir/LinuxAssignment/docs/file2.txt
/home/cdac/mydir/LinuxAssignment/file1.txt
/home/cdac/mydir/LinuxAssignment/input.txt
/home/cdac/output.txt
/home/cdac/output2.txt
/home/cdac/x1.txt
/home/cdac/x2.txt
/home/f1.txt
/home/f2.txt
/home/f3.txt
/home/f4.txt
/home/LinuxAssignment/d1.txt
/home/LinuxAssignment/data.txt
/home/LinuxAssignment/docs/file2.txt
/home/LinuxAssignment/file1.txt
/home/LinuxAssignment/input.txt
```

- `chmod u+x file.txt`

```
cdac@ADITYA-AMBAR:~$ ls -l x2.txt
-rw-r--r-- 1 cdac cdac 49 Aug 29 21:32 x2.txt
cdac@ADITYA-AMBAR:~$ chmod u+x x2.txt
cdac@ADITYA-AMBAR:~$ ls -l x2.txt
-rwxr--r-- 1 cdac cdac 49 Aug 29 21:32 x2.txt
```

- echo \$PATH

It displays the path of executable files

```
cdac@ADITYA-AMBAR:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/mnt/c/Program Files/Java/jdk-9/bin:/mnt/c/Users/aditya_Rangari/AppData/Local/Microsoft/WindowsApps:/mnt/d/Microsoft VS Code/bin:/mnt/c/Users/aditya_Rangari/AppData/Roaming/npm:/mnt/c/MinGW/bin:/snap/bin
cdac@ADITYA-AMBAR:~$
```

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

Identify the Incorrect Commands:

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

All are incorrect, the correct are below

1. chmod
2. cp
3. touch, nano, cat
4. cat
5. mv

Part C

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

echo "Hello, World"

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q1.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q1.sh
Hello, World!
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ |
```

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
```

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q2.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q2.sh
CDAC Mumbai
```

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

```
echo"Entered the Number"
```

```
read Num1
```

```
echo"The Entered Number is" $Num1
```

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q3.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q3.sh
Enter the Number
45
The Entered Number is 45
```

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

```
echo"Enter the first number: "
```

```
read n1
```

```
echo"Enter the second number: "
```

```
read n2
```

```
x=$((n1 + n2))
```

```
echo"The sum of $n1 and $n2 is: $x"
```

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q4.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q4.sh
Enter the first number:
5
Enter the second number:
3
The sum of 5 and 3 is: 8
```

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

```
echo"Enter the Number: "
```

```
read n1
```

```
if(( $n1%2==0 ))
```

```
then
```

```
    echo"The number $n1 is Even: "
```

else

echo "The number \$n1 is Odd: "

fi

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q5.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q5.sh
Enter the Number :
4
The number 4 is Even
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q5.sh
Enter the Number :
777777
The number is Odd
```

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

For ((i=1;i<=5;i++))

do

Echo Number : \$i

done

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q6.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q6.sh
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
```

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

i=1

while ((\$i<=5))

do

echo Number : \$i

((i++))

done

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q7.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q7.sh
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
```

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 "q7.sh"]

then

echo File exists

else

echo File does not exists

fi

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q8.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q8.sh
File exists
```

If [-f "Notfile.sh"]

then

echo File exists

else

echo File does not exists

fi

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q8.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q8.sh
File does not exist
```

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.

echo "Enter the number:"

read x

if((\$x>10))

then

echo "The number \$x is greater than 10"

else

echo "The number \$x is lesser than 10"

fi

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano q9.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q9.sh
Enter the number:
4
The number 4 is lesser than 10
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q9.sh
Enter the number:
54546542
The number 54546542 is greater than 10
```

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.

```
#!/bin/bash

for i in {1..5}
do
    for j in {1..5}
    do

        printf "%4d" $((i * j))

    done
    echo
done
```

```
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ sudo nano p10.sh
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash p10.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
```

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.

```
while true
do
    echo Enter the number
    read x

    if(( $x<0 ))
    then
        echo Negative number
        break
    else
        sq=$(( $x*$x ))
        echo Square is $sq
    fi
done
```

```

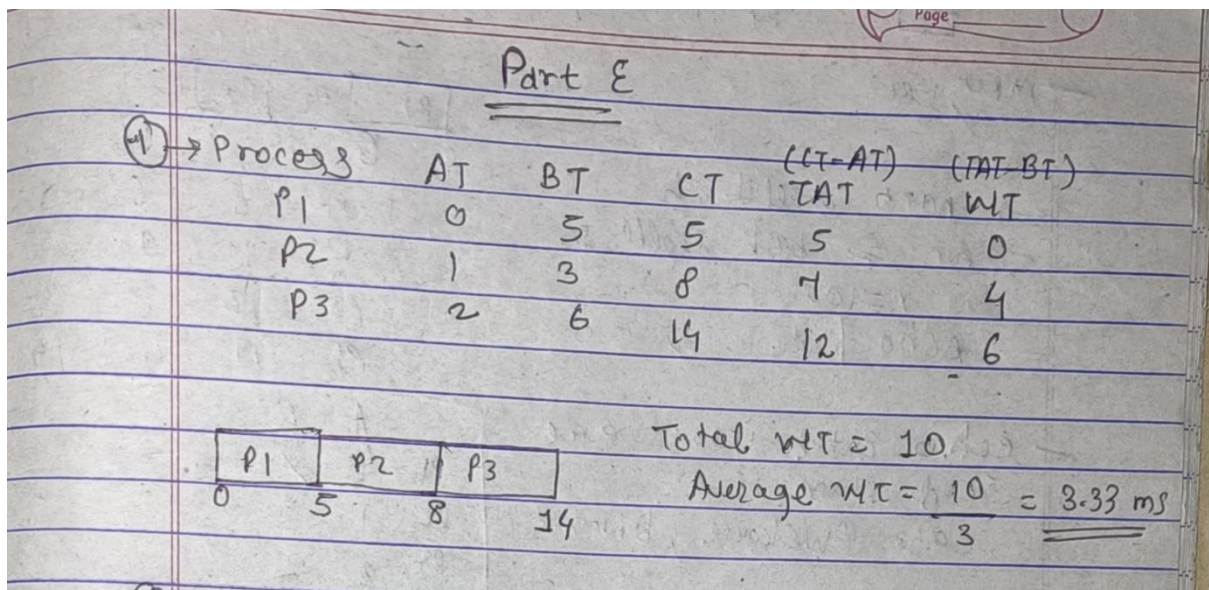
cdac@ADITYA-AMBAR:/home/LinuxAssignment/ShellProg$ bash q11.sh
Enter the number
8
Square is 64
Enter the number
7
Square is 49
Enter the number
-5
Negative number

```

1. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	6

Calculate the average waiting time using First-Come, First-Served (FCFS) scheduling.



2. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	3
P2	1	5
P3	2	1
P4	3	4

Calculate the average turnaround time using Shortest Job First (SJF) scheduling.

② →

Process	AT	BT	CT	(CT-AT) TAT
P1	0	3	3	3
P2	1	5	13	12
P3	2	1	4	2
P4	3	4	8	5

P1	P3	P4	P2
0	3	4	8
			13

Total TAT = 22
Average TAT = $\frac{22}{4} = 5.5$

3 Consider the following processes with arrival times, burst times, and priorities (lower number indicates higher priority):

| Process | Arrival Time | Burst Time | Priority |

|-----|-----|-----|-----|

| P1 | 0 | 6 | 3 |

| P2 | 1 | 4 | 1 |

| P3 | 2 | 7 | 4 |

| P4 | 3 | 2 | 2 |

Calculate the average waiting time using Priority Scheduling.

③ →

Process	AT	BT	Priority	CT	TAT	(TAT-BT) WT
P1	0	6	3	6	6	0
P2	1	4	1 (High)	10	9	5
P3	2	7	4 (Low)	19	17	10
P4	3	2	2	12	9	7

P1	P2	P4	P3
0	6	10	12
			19

Total WT = 22
Average WT = $\frac{22}{4} = 5.5$

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

| Process | Arrival Time | Burst Time |

|-----|-----|-----|

| P1 | 0 | 4 |

| P2 | 1 | 5 |

| P3 | 2 | 2 |

| P4 | 3 | 3 |

Calculate the average turnaround time using Round Robin scheduling.

Date _____
Page _____

$TQ = 2$

(4)

Process	AT	BT	CT	TAT
P1	0	4/2 ✓	8	8
P2	1	5/3 ✓	14	13
P3	2	2 ✓	6	4
P4	3	3/1 ✓	13	10

RQ: P1, P2, P3, P1, P4, P2, P4, P2

Total TAT = 35
Average TAT = 8.75

P1	P2	P3	P1	P4	P2	P4	P2
0	2	4	6	8	10	12	14

- 5 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?

(5)

Parent x = 5

fork

Child x

x = 6

Parent x

x = 6

final values of both is x = 6