


PART A


What will the following commands do?

- echo "Hello, World!" : → It display/print hello, world! On command prompt

 cdac@DESKTOP-903P6BA: ~


```
cdac@DESKTOP-903P6BA:~$ echo "hello, world!"
hello, world!
cdac@DESKTOP-903P6BA:~$
```

- name="Productive" :→ name is a variable which store the string

 cdac@DESKTOP-903P6BA: ~


```
cdac@DESKTOP-903P6BA:~$ name="Productive"
cdac@DESKTOP-903P6BA:~$ echo $name
Productive
cdac@DESKTOP-903P6BA:~$
```

- touch file.txt :→ It create a empty file

 cdac@DESKTOP-903P6BA: ~

```
cdac@DESKTOP-903P6BA:~$ touch file.txt
cdac@DESKTOP-903P6BA:~$ ls
Day_1          abc.txt  duplicate.txt  file1.txt  fruit.txt  numbers.txt
LinuxAssignment data.txt file.txt       file2.txt  input.txt
```

- ls -a :→ It list out all the files and directories including hidden files/directories

 cdac@DESKTOP-903P6BA: ~

```
cdac@DESKTOP-903P6BA:~$ ls -a
.  .bash_history  .bashrc  .local  .profile  Day_1  abc.txt  duplicate.txt  file1.txt  file644.txt  input.txt  numbers.txt
.. .bash_logout  .cache   .motd_shown  .sudo_as_admin_successful  LinuxAssignment  data.txt  file.txt  file2.txt  fruit.txt  mydir  script.sh
cdac@DESKTOP-903P6BA:~$
```

- rm file.txt :→ this delete the file.txt

```
cdac@DESKTOP-903P6BA:~$ rm file.txt
cdac@DESKTOP-903P6BA:~$ ls
Day_1 LinuxAssignment abc.txt data.txt duplicate.txt file1.txt file2.txt fruit.txt input.txt numbers.txt
cdac@DESKTOP-903P6BA:~$ cat file1.txt
```

- cp file1.txt file2.txt :→ it will copy the content of file1 into file2

```
cdac@DESKTOP-903P6BA:~$ cat file1.txt
Hello
It's a nice day
cdac@DESKTOP-903P6BA:~$ cat file2.txt
cdac@DESKTOP-903P6BA:~$ cp file1.txt file2.txt
cdac@DESKTOP-903P6BA:~$ cat file2.txt
Hello
It's a nice day
```


- mv file.txt /path/to/directory/ :→these moves the file into directory

```
cdac@DESKTOP-903P6BA:~$ mv file.txt /home/cdac/Day_1/
cdac@DESKTOP-903P6BA:~$ ls
Day_1  LinuxAssignment  abc.txt  data.txt  duplicate.txt  file1.txt  file2.txt  fruit.txt  input.txt  numbers.txt
cdac@DESKTOP-903P6BA:~$ cd Day_1
cdac@DESKTOP-903P6BA:~/Day_1$ ls
docs  file.txt
cdac@DESKTOP-903P6BA:~/Day_1$
```

- chmod 755 script.sh :→ it gives read,write and execute permission to the owner of the file


```
cdac@DESKTOP-903P6BA: ~
cdac@DESKTOP-903P6BA:~$ nano script.sh
cdac@DESKTOP-903P6BA:~$ ls -l
total 44
drwxr-xr-x 3 cdac cdac 4096 Aug 29 22:28 Day_1
drwxr-xr-x 3 cdac cdac 4096 Aug 29 19:36 LinuxAssignment
-rw-r--r-- 1 cdac cdac 263 Aug 28 22:19 abc.txt
-rw-r--r-- 1 cdac cdac 168 Aug 29 19:51 data.txt
-rw-r--r-- 1 cdac cdac 118 Aug 29 21:05 duplicate.txt
-rw-r--r-- 1 cdac cdac 23 Aug 29 19:40 file1.txt
-rwxr-x-- 1 cdac cdac 23 Aug 29 22:26 file2.txt
-rw-r--r-- 1 cdac cdac 176 Aug 29 21:13 fruit.txt
-rw-r--r-- 1 cdac cdac 44 Aug 29 20:56 input.txt
-rw-r--r-- 1 cdac cdac 71 Aug 29 19:57 numbers.txt
-rw-r--r-- 1 cdac cdac 19 Aug 29 22:34 script.sh
cdac@DESKTOP-903P6BA:~$ chmod 755 script.sh
cdac@DESKTOP-903P6BA:~$ ls -l
total 44
drwxr-xr-x 3 cdac cdac 4096 Aug 29 22:28 Day_1
drwxr-xr-x 3 cdac cdac 4096 Aug 29 19:36 LinuxAssignment
-rw-r--r-- 1 cdac cdac 263 Aug 28 22:19 abc.txt
-rw-r--r-- 1 cdac cdac 168 Aug 29 19:51 data.txt
-rw-r--r-- 1 cdac cdac 118 Aug 29 21:05 duplicate.txt
-rw-r--r-- 1 cdac cdac 23 Aug 29 19:40 file1.txt
-rwxr-x-- 1 cdac cdac 23 Aug 29 22:26 file2.txt
-rw-r--r-- 1 cdac cdac 176 Aug 29 21:13 fruit.txt
-rw-r--r-- 1 cdac cdac 44 Aug 29 20:56 input.txt
-rw-r--r-- 1 cdac cdac 71 Aug 29 19:57 numbers.txt
-rwxr-xr-x 1 cdac cdac 19 Aug 29 22:34 script.sh
cdac@DESKTOP-903P6BA:~$
```

- `grep "pattern" file.txt` :→ it searches for pattern specific word in the file


 `cdac@DESKTOP-903P6BA: ~`

```
cdac@DESKTOP-903P6BA:~$ nano file.txt
cdac@DESKTOP-903P6BA:~$ cat file.txt
hi
hey
pattern
good day
cdac@DESKTOP-903P6BA:~$ grep "pattern" file.txt
pattern
cdac@DESKTOP-903P6BA:~$
```

- kill PID
- `mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt` :-
 1. it will first create mydir directory
 2. touch command create file.txt file in mydir directory
 3. > redirection command copy hello world in file.txt
 4. Cat command display the content in the file.txt

```
 cdac@DESKTOP-903P6BA: ~/mydir
cdac@DESKTOP-903P6BA:~$ mkdir mydir && cd mydir && touch files.txt && echo "Hello, World!" > files.txt && cat files.txt
Hello, World!
cdac@DESKTOP-903P6BA:~/mydir$ cd ..
cdac@DESKTOP-903P6BA:~$ ls
Day_1 LinuxAssignment abc.txt data.txt duplicate.txt file.txt file1.txt file2.txt fruit.txt input.txt mydir numbers.txt script.sh
cdac@DESKTOP-903P6BA:~$ cd mydir
cdac@DESKTOP-903P6BA:~/mydir$ ls
files.txt
cdac@DESKTOP-903P6BA:~/mydir$
```

- `ls -l | grep ".txt"` :→ it list out all the ".txt" files in directories

```
 cdac@DESKTOP-903P6BA: ~/mydir
cdac@DESKTOP-903P6BA:~/mydir$ ls -l | grep ".txt"
-rw-r--r-- 1 cdac cdac 14 Aug 29 22:43 files.txt
cdac@DESKTOP-903P6BA:~/mydir$
```

- `cat file1.txt file2.txt | sort | uniq` :→ it show the content of file1 and file 2 also sorted out content

And print only unique content

```
cdac@DESKTOP-903P6BA: ~  
cdac@DESKTOP-903P6BA:~$ cat file1.txt  
Hello  
It's a nice day  
hi  
it,s a nice day  
hi  
hi  
who's there?  
knock knock  
you are beautiful  
  
cdac@DESKTOP-903P6BA:~$ cat file1.txt file2.txt |sort|uniq  
Hello  
It's a nice day  
hi  
it,s a nice day  
knock knock  
who's there?  
you are beautiful  
cdac@DESKTOP-903P6BA:~$
```

- `ls -l | grep "^d"` :→it list out all the directories

```
cdac@DESKTOP-903P6BA:~$  
cdac@DESKTOP-903P6BA:~$ ls -l | grep "^d"  
drwxr-xr-x 3 cdac cdac 4096 Aug 29 22:28 Day_1  
drwxr-xr-x 3 cdac cdac 4096 Aug 29 19:36 LinuxAssignment  
drwxr-xr-x 2 cdac cdac 4096 Aug 29 22:56 mydir  
cdac@DESKTOP-903P6BA:~$
```

- `grep -r "pattern" /path/to/directory/` :→ it recursively search in all the files of directory to

Match the “pattern” string

```
cdac@DESKTOP-903P6BA:~$ grep -r "pattern" /home/cdac/file.txt  
pattern  
cdac@DESKTOP-903P6BA:~$
```

- `cat file1.txt file2.txt | sort | uniq -d` :→ it concatenate file1 and file2 and sort the content in
Both the file and uniq command list the lines and repeated line only printed once

```
cdac@DESKTOP-903P6BA: ~
cdac@DESKTOP-903P6BA:~$ cat file1.txt
Hello
It's a nice day
hi
it,s a nice day
hi
hi
who's there?
knock knock
you are beautiful

cdac@DESKTOP-903P6BA:~$ cat file1.txt file2.txt |sort|uniq -d
Hello
It's a nice day
hi
cdac@DESKTOP-903P6BA:~$
```

- `chmod 644 file.txt` :→ these give read write permission to the user and execute permission to
The group and others

```
cdac@DESKTOP-903P6BA:~$ chmod 644 file644.txt
cdac@DESKTOP-903P6BA:~$ ls -l
total 52
drwxr-xr-x 3 cdac cdac 4096 Aug 29 22:28 Day_1
drwxr-xr-x 3 cdac cdac 4096 Aug 29 19:36 LinuxAssignment
-rw-r--r-- 1 cdac cdac 263 Aug 28 22:19 abc.txt
-rw-r--r-- 1 cdac cdac 168 Aug 29 19:51 data.txt
-rw-r--r-- 1 cdac cdac 118 Aug 29 21:05 duplicate.txt
-rw-r--r-- 1 cdac cdac 0 Aug 29 23:31 file.txt
-rw-r--r-- 1 cdac cdac 92 Aug 29 23:02 file1.txt
-rwxr-x--- 1 cdac cdac 23 Aug 29 22:26 file2.txt
-rw-r--r-- 1 cdac cdac 10 Aug 29 23:32 file644.txt
-rw-r--r-- 1 cdac cdac 176 Aug 29 21:13 fruit.txt
-rw-r--r-- 1 cdac cdac 44 Aug 29 20:56 input.txt
drwxr-xr-x 2 cdac cdac 4096 Aug 29 22:56 mydir
-rw-r--r-- 1 cdac cdac 71 Aug 29 19:57 numbers.txt
-rwxr-xr-x 1 cdac cdac 19 Aug 29 22:34 script.sh
cdac@DESKTOP-903P6BA:~$
```

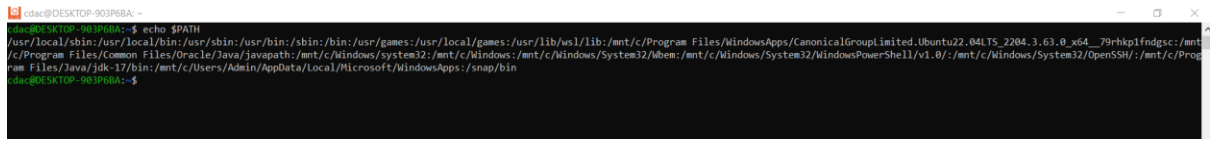
- `cp -r source_directory destination_directory` :→ these copy all contents from a source directory to the destination directory

```
cdac@DESKTOP-903P6BA:~$ cp -r abc.txt data.txt
cdac@DESKTOP-903P6BA:~$ cat data.txt
hi
hey
hello
what's up?
heyyy
good morning
good afternoon
good evening
how do you do?
it's nice to meet you
it's a pleasure to meet you
welcome
how's it going?
how are you?
long time no see
what's new?
who's there?
hey beautiful
knock knock..
how is life going?
cdac@DESKTOP-903P6BA:~$
```

- `find /path/to/search -name "*.txt"`
- `chmod u+x file.txt` :→ it gives permission to user to execute file

```
cdac@DESKTOP-903P6BA:~$ chmod u+x file.txt
cdac@DESKTOP-903P6BA:~$ ls -l
total 52
drwxr-xr-x 3 cdac cdac 4096 Aug 29 22:28 Day_1
drwxr-xr-x 3 cdac cdac 4096 Aug 29 19:36 LinuxAssignment
-rw-r--r-- 1 cdac cdac 263 Aug 28 22:19 abc.txt
-rw-r--r-- 1 cdac cdac 263 Aug 29 23:38 data.txt
-rw-r--r-- 1 cdac cdac 118 Aug 29 21:05 duplicate.txt
-rwxr--r-- 1 cdac cdac  0 Aug 29 23:31 file.txt
-rw-r--r-- 1 cdac cdac  92 Aug 29 23:02 file1.txt
-rwxr-x--- 1 cdac cdac  23 Aug 29 22:26 file2.txt
-rw-r--r-- 1 cdac cdac  10 Aug 29 23:32 file644.txt
-rw-r--r-- 1 cdac cdac 176 Aug 29 21:13 fruit.txt
-rw-r--r-- 1 cdac cdac  44 Aug 29 20:56 input.txt
drwxr-xr-x 2 cdac cdac 4096 Aug 29 22:56 mydir
-rw-r--r-- 1 cdac cdac  71 Aug 29 19:57 numbers.txt
-rwxr-xr-x 1 cdac cdac  19 Aug 29 22:34 script.sh
cdac@DESKTOP-903P6BA:~$
```

- **echo \$PATH** :→ it allows us to view the current value of the \$PATH variable in a linux system



```
cdac@DESKTOP-903P6BA:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/ssl/lib:/mnt/c/Program Files/WindowsApps/CanonicalGroupLimited.Ubuntu22.04 LTS_2204.3.63.0_x64__79rhkp1fndgsc:/mnt/c/Program Files/Common Files/Oracle/Java/javapath:/mnt/c/windows/system32:/mnt/c/windows:/mnt/c/windows/system32/wbem:/mnt/c/windows/system32/windowspowershell/v1.0:/mnt/c/windows/system32/openssh:/mnt/c/Program Files/Java/jdk-17/bin:/mnt/c/Users/Admin/AppData/Local/Microsoft/WindowsApps:/snap/bin
cdac@DESKTOP-903P6BA:~$
```

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. → **True**
2. **cpy** is used to copy files and directories. → **False**
3. **mkfile** is used to create a new file. → **True**
4. **catx** is used to concatenate files. → **False**
5. **rn** is used to rename files. → **True**

PART C

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

```
cdac@DESKTOP-903P6BA:~$  
cdac@DESKTOP-903P6BA:~$ echo "Hello,world!"  
Hello,world!  
cdac@DESKTOP-903P6BA:~$
```

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

```
cdac@DESKTOP-903P6BA: ~  
cdac@DESKTOP-903P6BA:~$ name="CDAC Mumbai"  
cdac@DESKTOP-903P6BA:~$ echo $name  
CDAC Mumbai
```

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

<pre>cdac@DESKTOP-903P6BA: ~ cdac@DESKTOP-903P6BA:~\$ nano p1 cdac@DESKTOP-903P6BA:~\$ bash p1 Enter the Number 25 number:- 25 cdac@DESKTOP-903P6BA:~\$</pre>	<pre>cdac@DESKTOP-903P6BA: ~ GNU nano 6.2 echo Enter the Number1 read num1 echo Number :- \$num1</pre>
---	--

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

<pre>cdac@DESKTOP-903P6BA: ~ cdac@DESKTOP-903P6BA:~\$ nano p1 cdac@DESKTOP-903P6BA:~\$ bash p1 Enter the Number1 5 Enter the Number2 3 result :- 8 cdac@DESKTOP-903P6BA:~\$</pre>	<pre>cdac@DESKTOP-903P6BA: ~ GNU nano 6.2 echo Enter the Number1 read num1 echo Enter the Number2 read num2 result=\$((num1 + num2)) echo result :- \$result</pre>
---	--

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

<pre>cdac@DESKTOP-903P6BA: ~ cdac@DESKTOP-903P6BA:~\$ nano p1.sh cdac@DESKTOP-903P6BA:~\$ bash p1.sh enter the number 2 2 is even cdac@DESKTOP-903P6BA:~\$ bash p1.sh enter the number 3 3 is odd cdac@DESKTOP-903P6BA:~\$</pre>	<pre>cdac@DESKTOP-903P6BA: ~ GNU nano 6.2 echo "enter the number" read num r=\$((\$num % 2)) if [\$r -eq 0] then echo "\$num is even" else echo "\$num is odd" fi</pre>
--	---

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

<pre>cdac@DESKTOP-903P6BA: ~ cdac@DESKTOP-903P6BA:~\$ nano for1_5.sh cdac@DESKTOP-903P6BA:~\$ bash for1_5.sh 1 2 3 4 5 cdac@DESKTOP-903P6BA:~\$</pre>	<pre>cdac@DESKTOP-903P6BA: ~ GNU nano 6.2 for((i=1;i<=5;i++)) do echo \$i done</pre>
---	---

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

<pre>cdac@DESKTOP-903P6BA: ~ cdac@DESKTOP-903P6BA:~\$ nano while1_5.sh cdac@DESKTOP-903P6BA:~\$ bash while1_5.sh 1 2 3 4 5 cdac@DESKTOP-903P6BA:~\$</pre>	<pre>cdac@DESKTOP-903P6BA: ~ GNU nano 6.2 i=1 while [\$i -le 5] do echo \$i i=`expr \$i + 1` done</pre>
---	---

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".

```
cdac@DESKTOP-903P6BA: ~
cdac@DESKTOP-903P6BA:~$ nano filecheck.sh
cdac@DESKTOP-903P6BA:~$ bash filecheck.sh
File does not exists
cdac@DESKTOP-903P6BA:~$ nano filecheck.sh
cdac@DESKTOP-903P6BA:~$ bash filecheck.sh
File exists
cdac@DESKTOP-903P6BA:~$
```

```
cdac@DESKTOP-903P6BA: ~
GNU nano 6.2
if [ -f /home/cdac/file1.txt ]
then
echo "File exists"
else
echo "File does not exists"
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.

```
cdac@DESKTOP-903P6BA: ~
cdac@DESKTOP-903P6BA:~$ nano p1.sh
cdac@DESKTOP-903P6BA:~$ bash p1.sh
enter the number
15
15 is greater than 10
cdac@DESKTOP-903P6BA:~$
```

```
cdac@DESKTOP-903P6BA: ~
GNU nano 6.2
echo "enter the number"
read num
if [ $num -gt 10 ]
then
echo "$num is greater than 10"
else
echo "$num is not greater than 10"
fi
```

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.

```
cdac@DESKTOP-903P6BA: ~
cdac@DESKTOP-903P6BA:~$ nano nesforMul.sh
cdac@DESKTOP-903P6BA:~$ bash nesforMul.sh
```

1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25
6	6	12	18	24	30
7	7	14	21	28	35
8	8	16	24	32	40
9	9	18	27	36	45
10	10	20	30	40	50

```
cdac@DESKTOP-903P6BA:~$
```

```
cdac@DESKTOP-903P6BA: ~
GNU nano 6.2
echo -----+-----
for a in {1..10}
do
echo -n "$a | "
for b in {1..5}
do
echo -en "\t$(($a * $b ))"
done
echo
done
echo -----+-----
```

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.

cdac@DESKTOP-903P6BA: ~

GNU nano 6.2

```
echo enter the number
read num1
while [ $num1 -lt 0 ]
do
echo enter the number
read num1
if [ $num1 -gt 0 ]
then
sq=`expr $num1 \* $num1`
echo square = $sq
else
break
fi
done
```

cdac@DESKTOP-903P6BA: ~

cdac@DESKTOP-903P6BA:~\$ bash positiveneg.sh

enter the number

-3

enter the number

8

square = 64

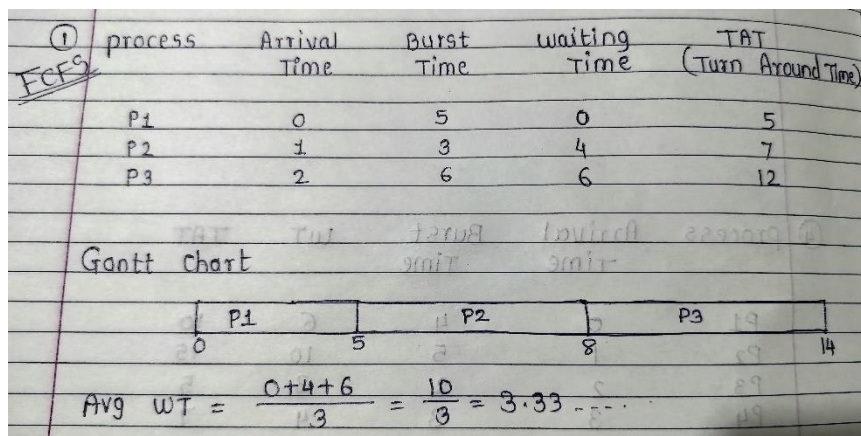
cdac@DESKTOP-903P6BA:~\$

PART E

- Consider the following processes with arrival times and burst times:

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

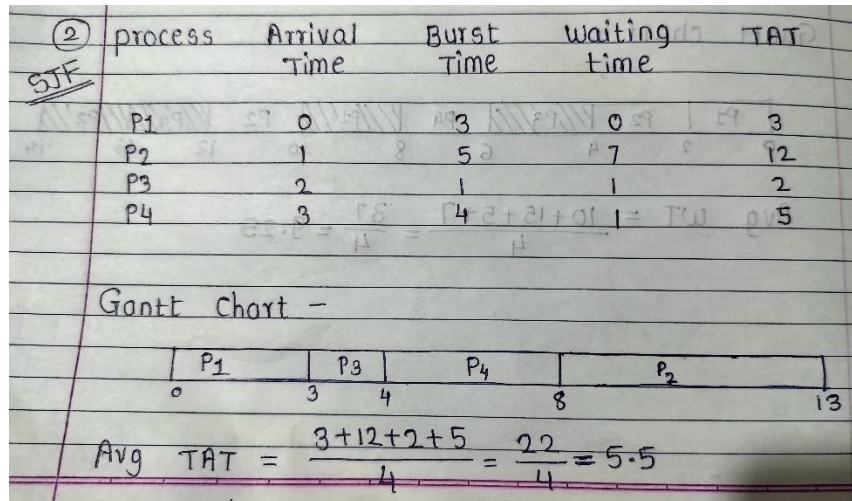
Calculate the average waiting time using First-Come, First-Served (FCFS) scheduling.



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

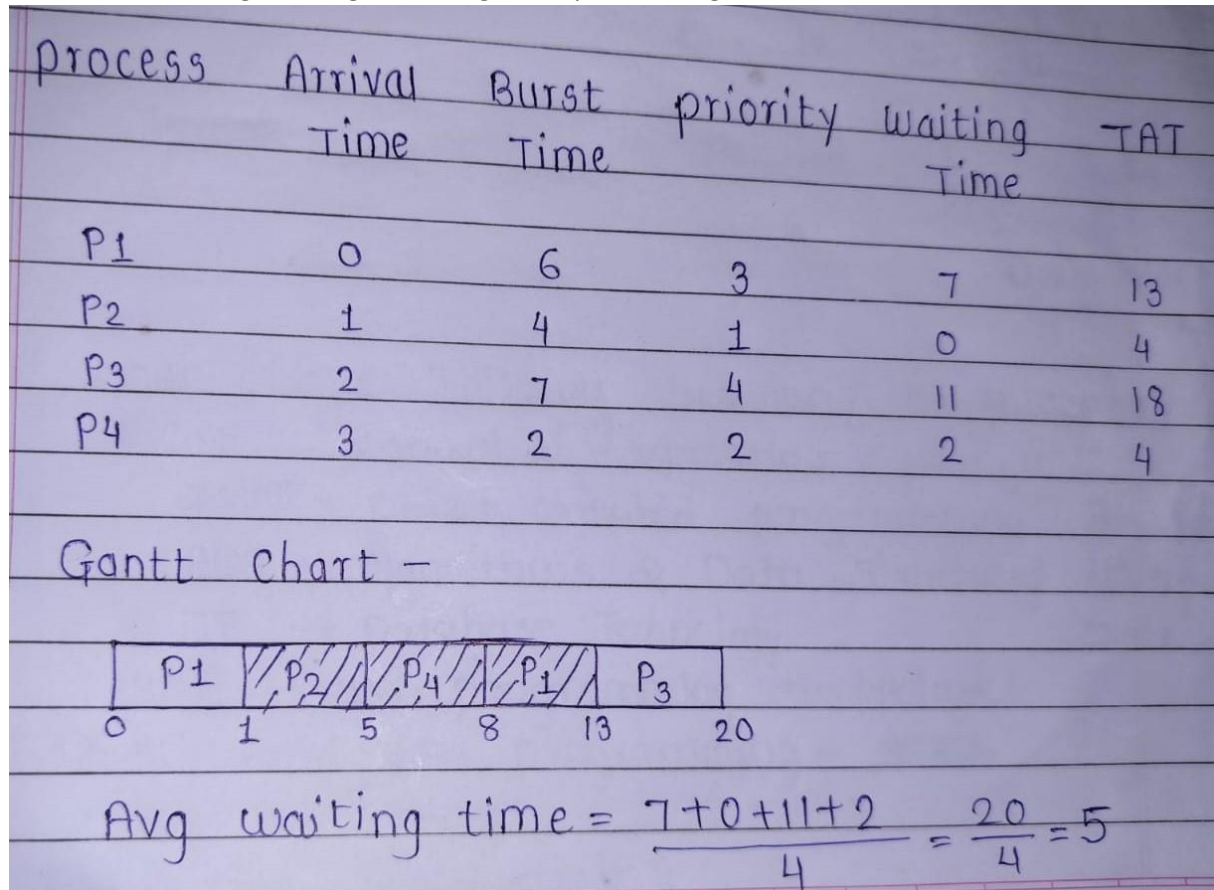
Calculate the average turnaround time using Shortest Job First (SJF) scheduling.



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.



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.

