**COS Assignment 2**

**Part A**

1. echo "Hello, World!" – Print Hello, World!
2. name="Productive" – assign Productive to variable name
3. touch file.txt – create new file.txt
4. ls -a – list all contents in directory including hidden files
5. rm file.txt – remove file.txt from directory
6. cp file1.txt file2.txt – copy content of file1.txt to file2.txt
7. mv file.txt /path/to/directory/ - move the file.txt to directory
8. chmod 755 script.sh – change file permission read, write, execute for owner, read and execute to group and other users
9. grep "pattern" file.txt – return all lines in file1.txt that include pattern
10. kill PID – terminate process with respected PID
11. mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!"> file.txt && cat file.txt – create mydir directory and move in it , create empty file file.txt and print the content of file1.txt
12. ls -l | grep ".txt" – print the list and find and show the line contains .txt
13. cat file1.txt file2.txt | sort | uniq – concatenate file1.txt and file2.txt and sort in alphabetical order and list unique entries
14. ls -l | grep "^d" - lists directories in long format and filters only lines that start with d
15. grep -r "pattern" /path/to/directory/ - search for the pattern in /path/to/directory/ directory
16. cat file1.txt file2.txt | sort | uniq -d - concatenate file1.txt and file2.txt and sort them and show only duplicate lines
17. chmod 644 file.txt – file settings change to read, write to owner , read group, read other users.
18. cp -r source\_directory destination\_directory – copy source\_directory recursively to destination\_directory
19. find /path/to/search -name "\*.txt" – find all .txt files in directory
20. chmod u+x file.txt – change file permission to allow owner to execute
21. echo $PATH – it will print list of colon-separated executable files

**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. -True

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.

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

**Identify the Incorrect Commands:**

1. chmodx is used to change file permissions. – chmod +x

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 file.txt file2.txt

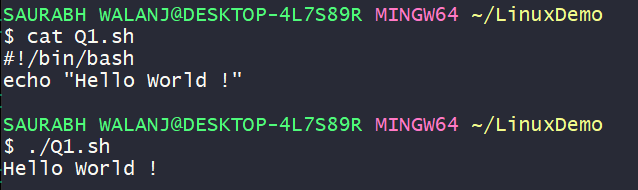
5. rn is used to rename files. – mv

**Part C**

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

#!/bin/bash

echo "Hello World !"

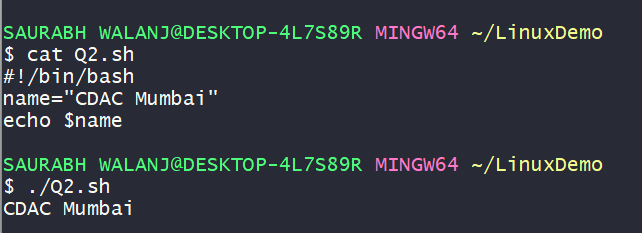


**Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable**.

#!/bin/bash

name="CDAC Mumbai"

echo $name



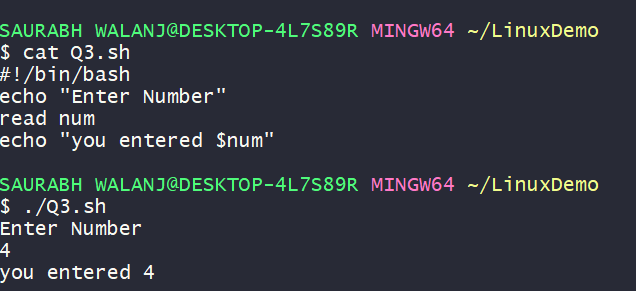
**Question 3: Write a shell script that takes a number as input from the user and prints it.**

#!/bin/bash

echo "Enter Number"

read num

echo "you entered $num"



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

echo "Enter 1st num"

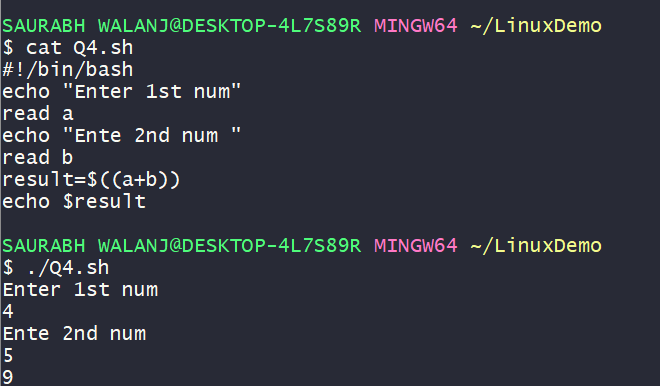
read a

echo "Enter 2nd num"

read b

result=$((a + b))

echo "Result = $result"



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

#!/bin/bash

echo "Enter the Number"

read x

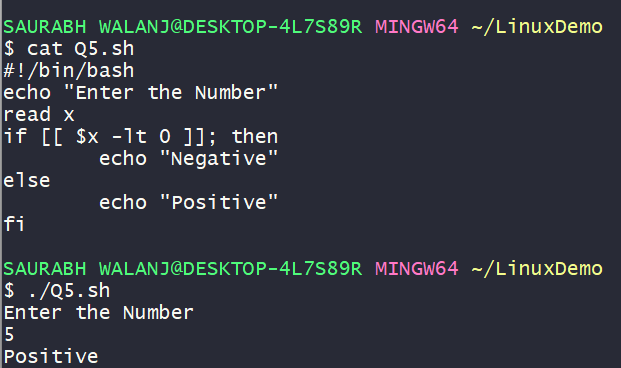
if [[ $x -lt 0 ]]; then

echo "Negative"

else

echo "Positive"

fi



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

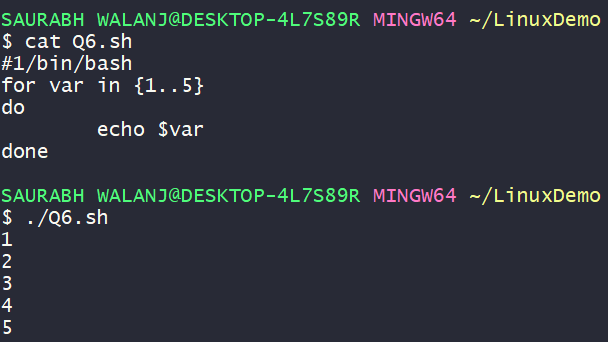
#1/bin/bash

for var in {1..5}

do

echo $var

done



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

$ cat Q7.sh

#!/bin/bash

i=1

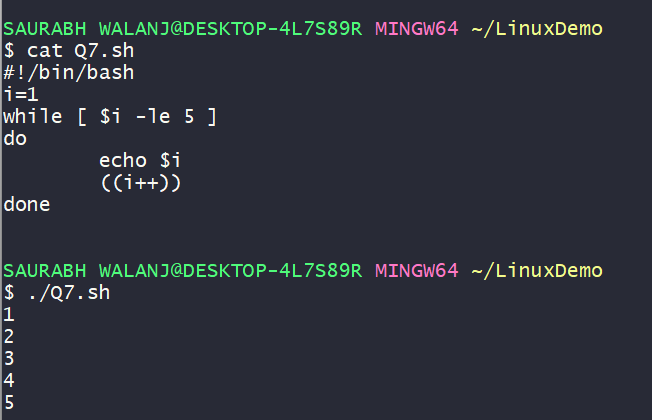
while [ $i -le 5 ]

do

echo $i

((i++))

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".**

#!/bin/bash

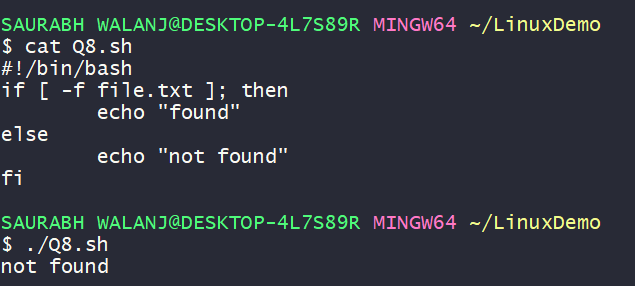
if [ -f file.txt ]; then

echo "found"

else

echo "not found"

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.**

#!/bin/bash

echo "enter the number"

read num

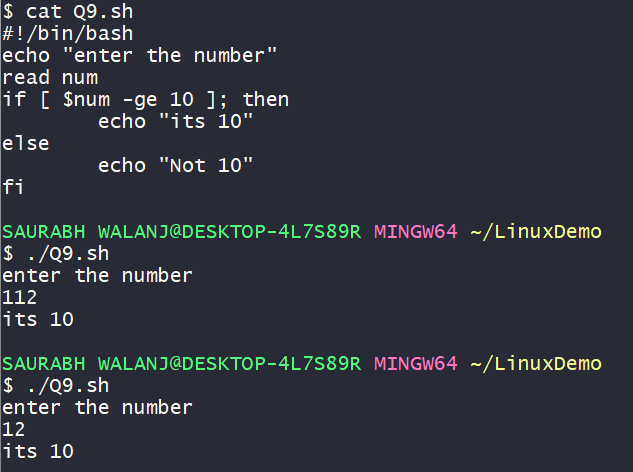
if [ $num -ge 10 ]; then

echo "its 10"

else

echo "Not 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.**

#!/bin/bash

for i in {1..5}

do

for j in {1..10}

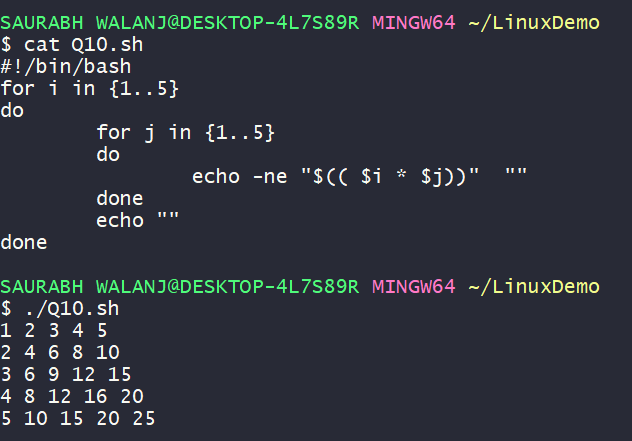
do

echo -ne "$((i \* j))" \t

done

echo ""

done



**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.**

#!/bin/bash

while true

do

echo "Enter the Number"

read i

if [[ $i -gt 0 ]]; then

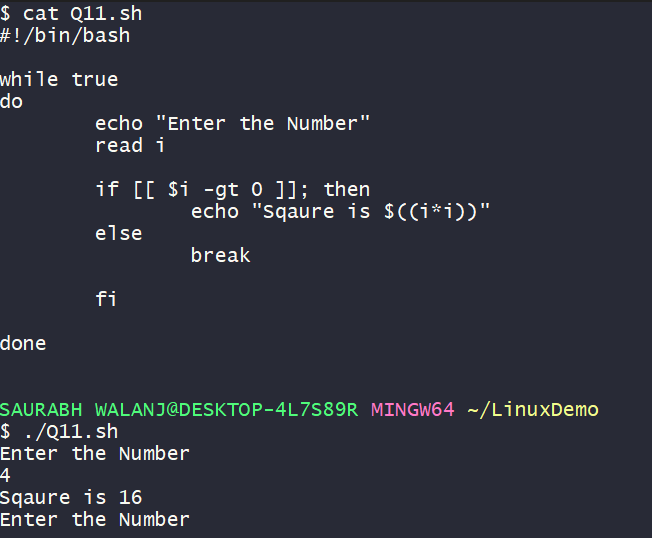
echo "Square is $((i\*i))"

else

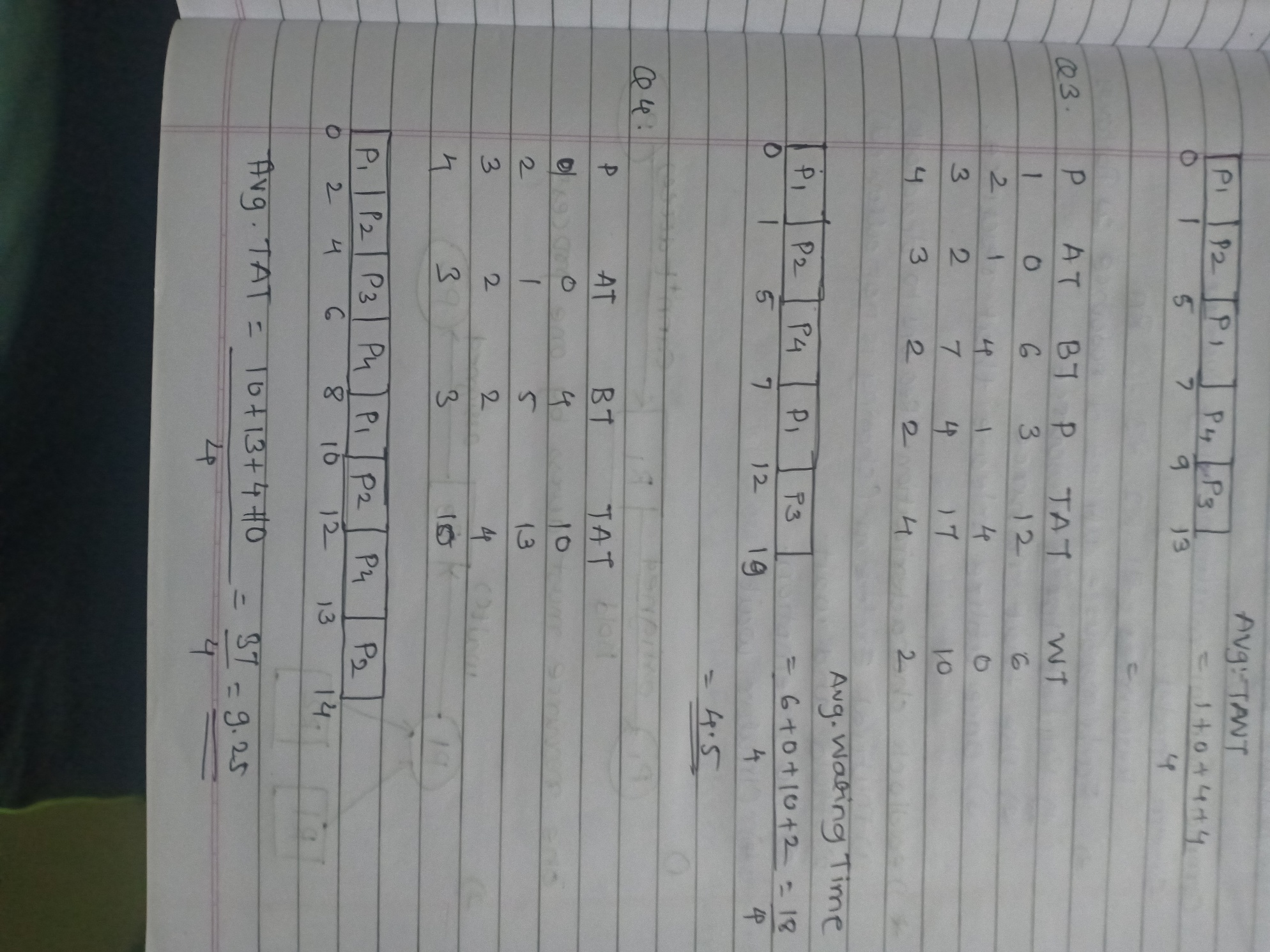
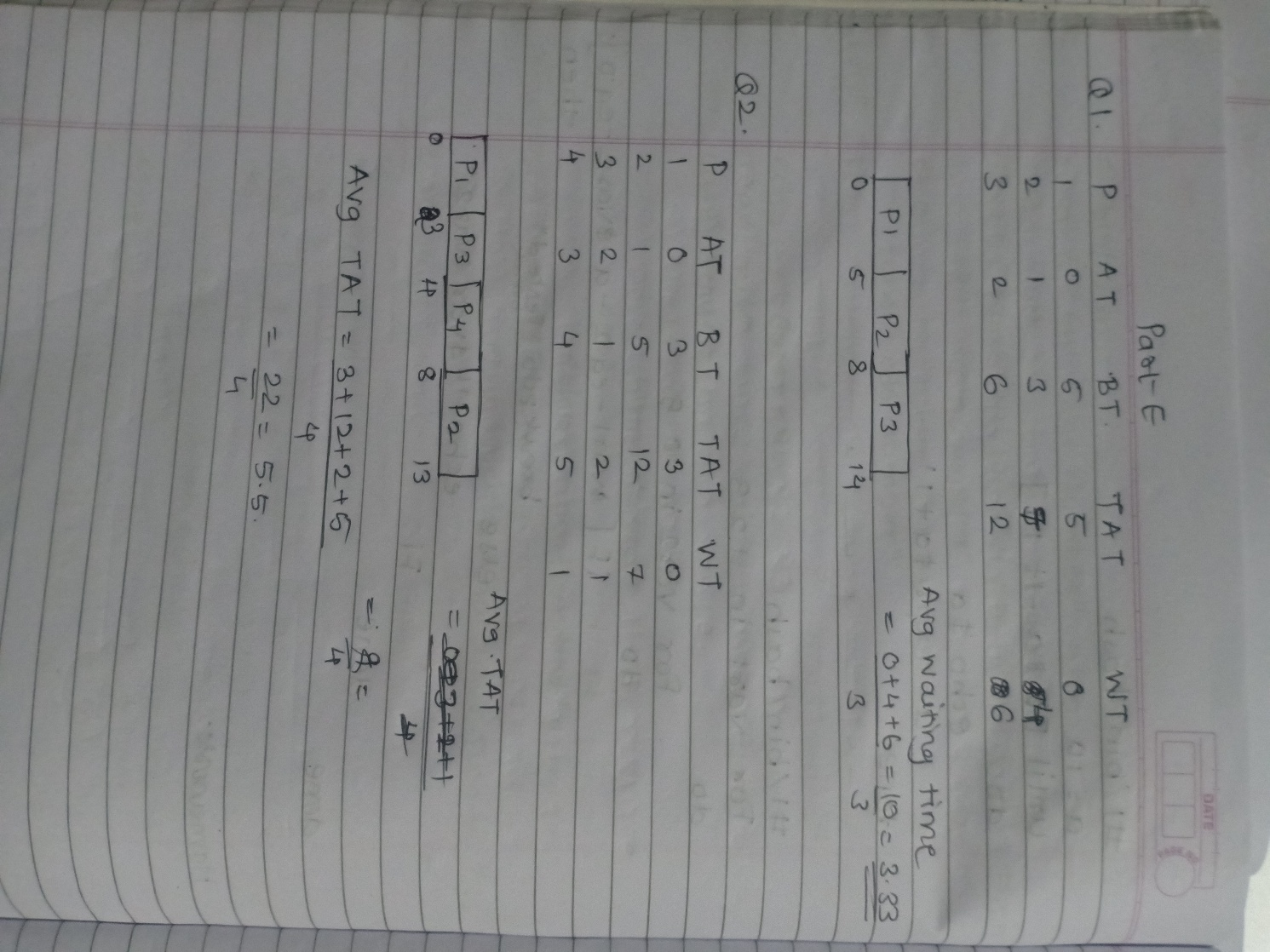
break

fi

done



**Part E**



**Q5**

After fork (), both parent and child get separate copies with values of x = 5. Both parent and child increment their x by 1.so final value is 6 in parent and 6 in child.