Assignment -2

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 (cp is used for copy; cd is used to change directory)

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. - chmod

2. cpy is used to copy files and directories. - cp

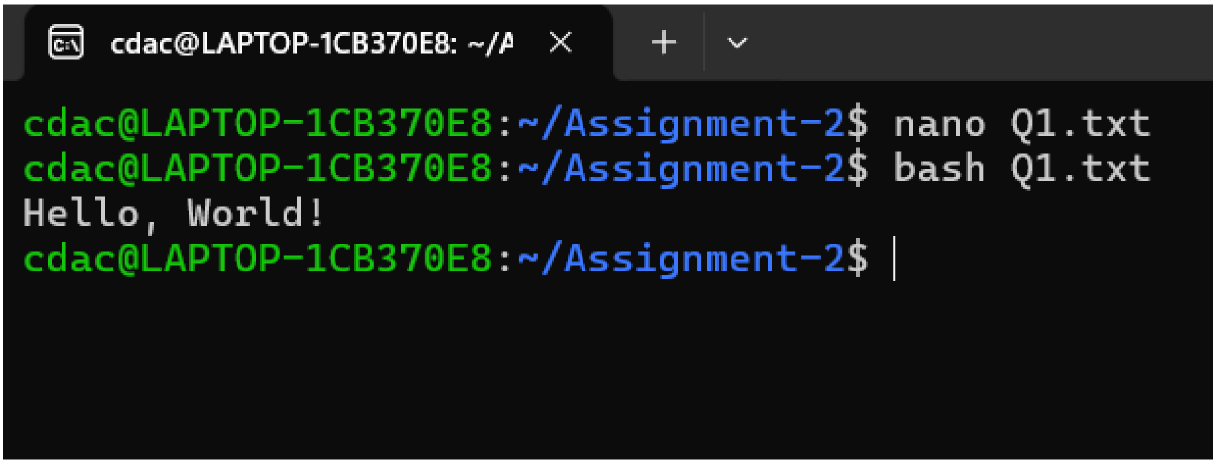
3. mkfile is used to create a new file. – touch/nano

4. catx is used to concatenate files. -cat

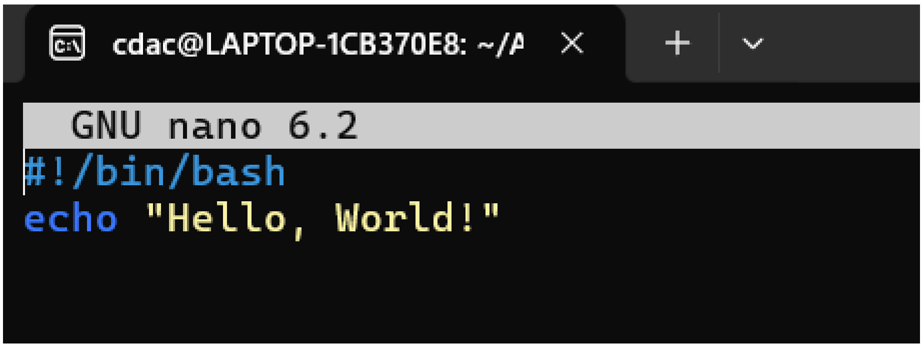
5. rn is used to rename files. - re

Part C

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

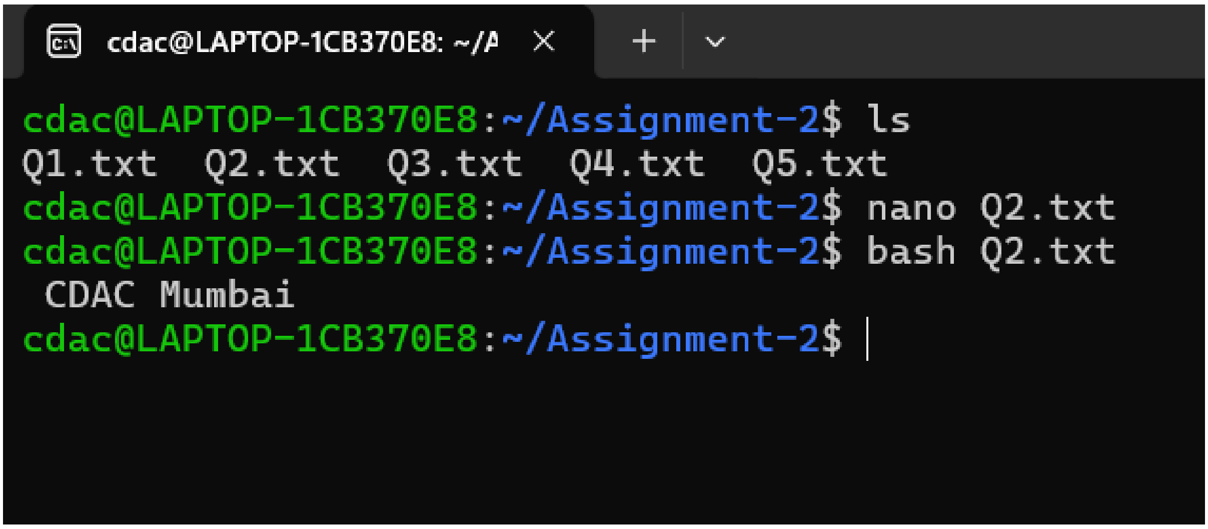


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q1.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q1.txt
Hello, World!
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

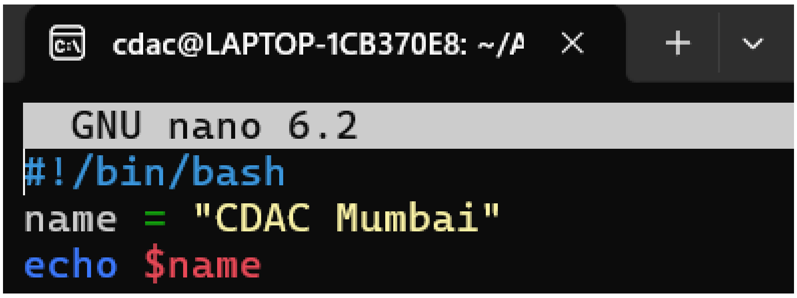


```
  GNU nano 6.2
#!/bin/bash
echo "Hello, World!"
```

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

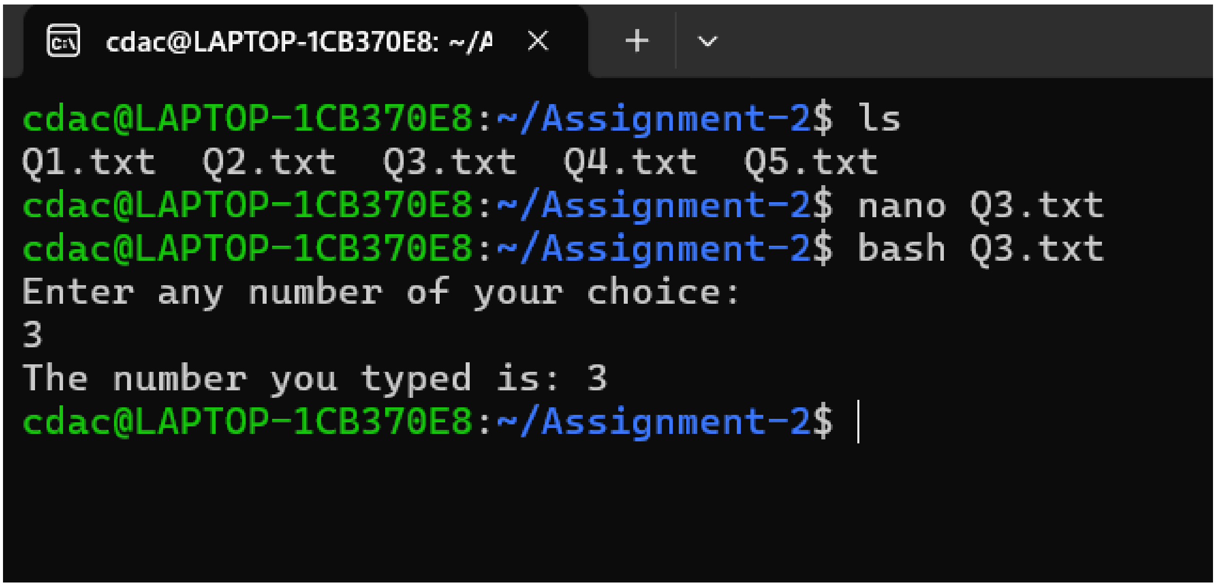


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q2.txt  Q3.txt  Q4.txt  Q5.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q2.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q2.txt
 CDAC Mumbai
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

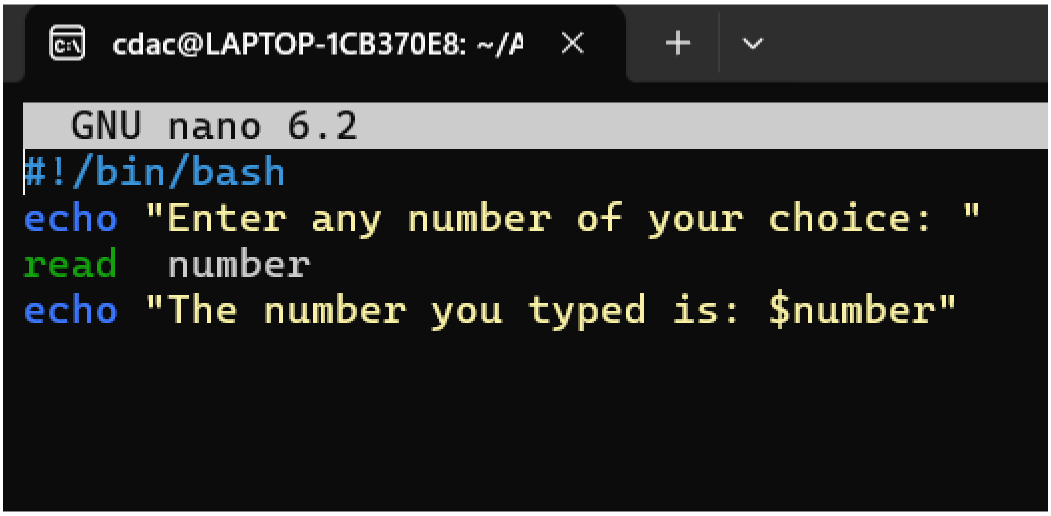


```
  GNU nano 6.2
#!/bin/bash
name = "CDAC Mumbai"
echo $name
```

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

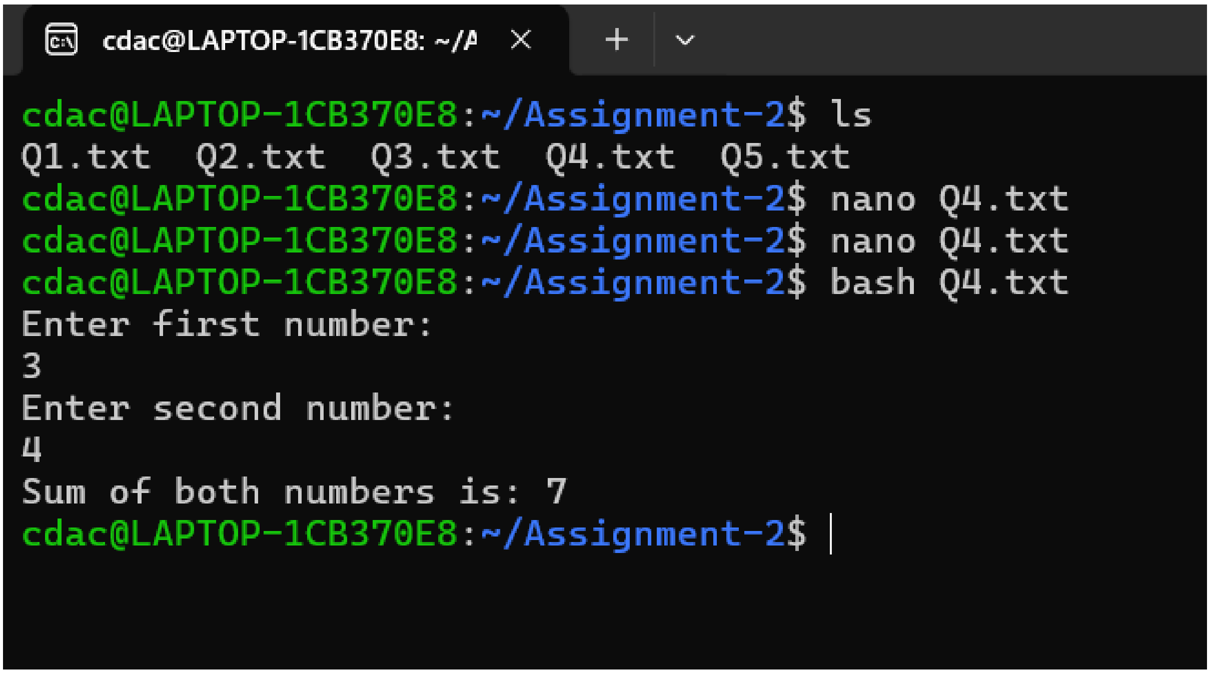


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q2.txt  Q3.txt  Q4.txt  Q5.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q3.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q3.txt
Enter any number of your choice:
3
The number you typed is: 3
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

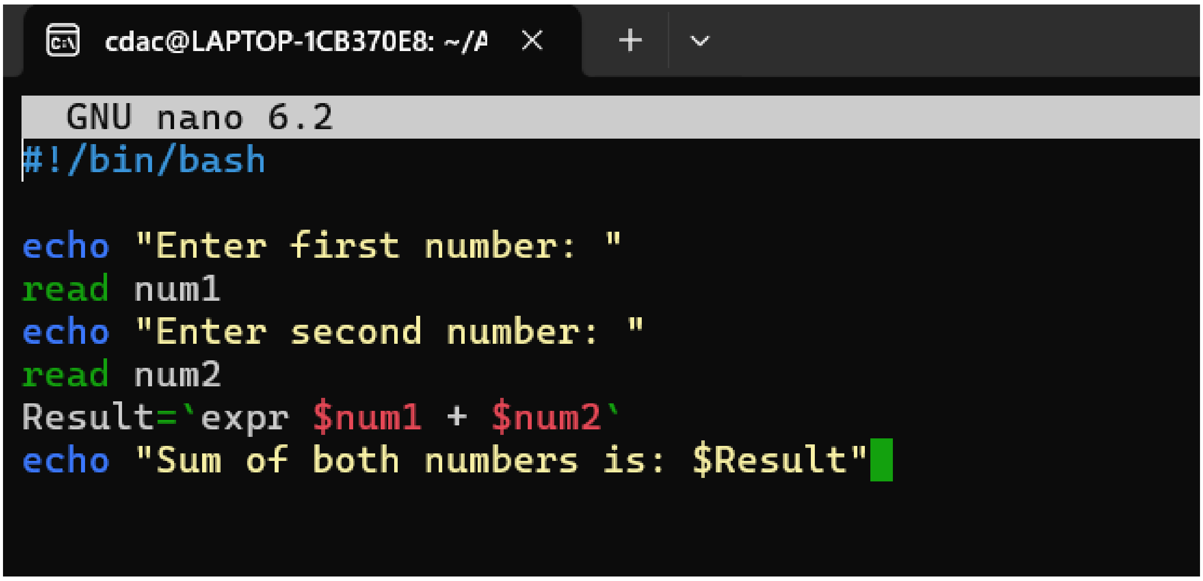


```
  GNU nano 6.2
#!/bin/bash
echo "Enter any number of your choice: "
read  number
echo "The number you typed is: $number"
```

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

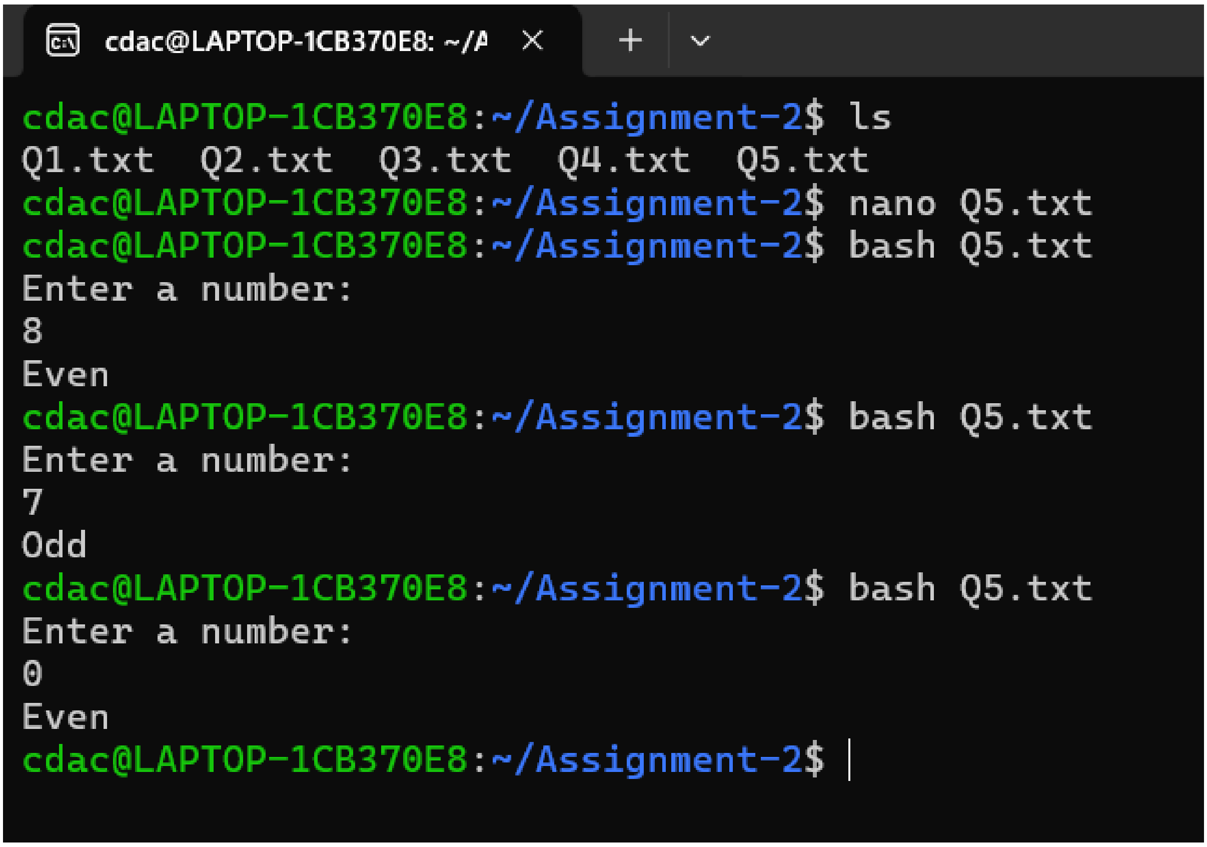


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q2.txt  Q3.txt  Q4.txt  Q5.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q4.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q4.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q4.txt
Enter first number:
3
Enter second number:
4
Sum of both numbers is: 7
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

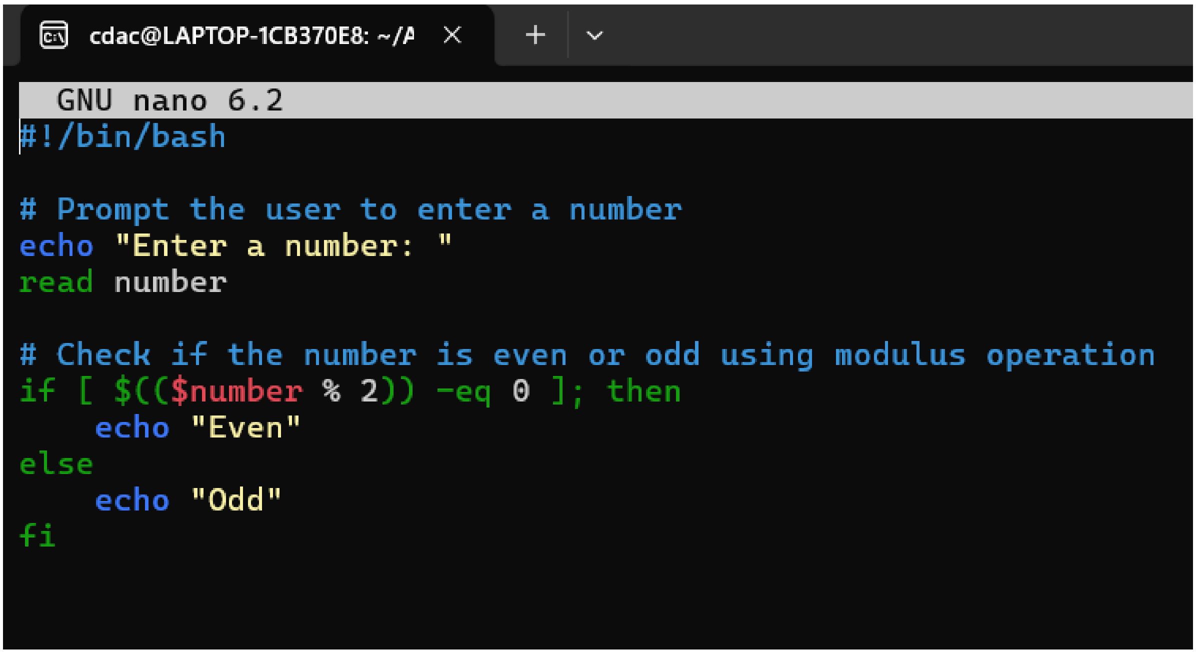


```
  GNU nano 6.2
#!/bin/bash

echo "Enter first number: "
read num1
echo "Enter second number: "
read num2
Result=`expr $num1 + $num2`
echo "Sum of both numbers is: $Result"
```

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

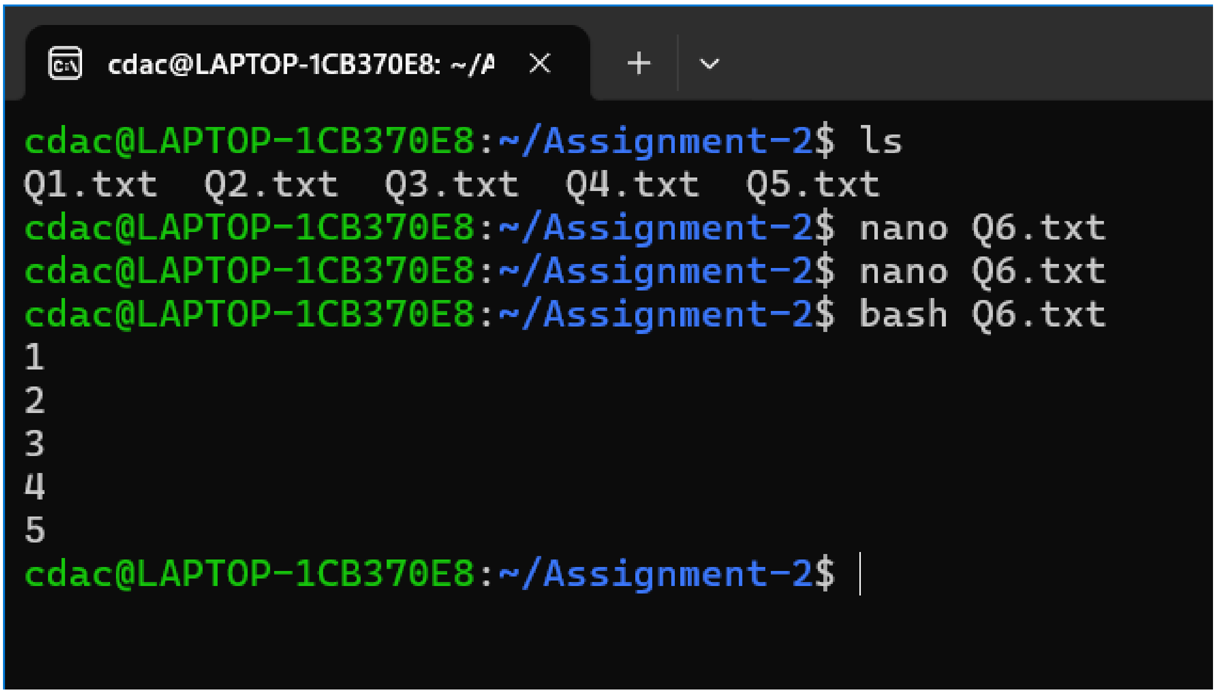


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q2.txt  Q3.txt  Q4.txt  Q5.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q5.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q5.txt
Enter a number:
8
Even
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q5.txt
Enter a number:
7
Odd
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q5.txt
Enter a number:
0
Even
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

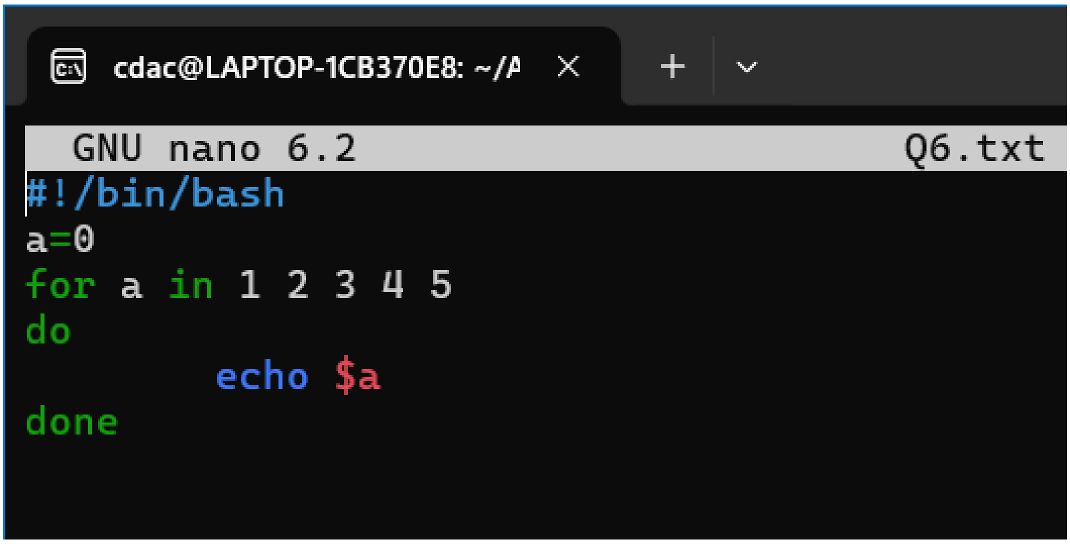


```
  GNU nano 6.2
#!/bin/bash

# Prompt the user to enter a number
echo "Enter a number: "
read number

# Check if the number is even or odd using modulus operation
if [ $(($number % 2)) -eq 0 ]; then
    echo "Even"
else
    echo "Odd"
fi
```

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

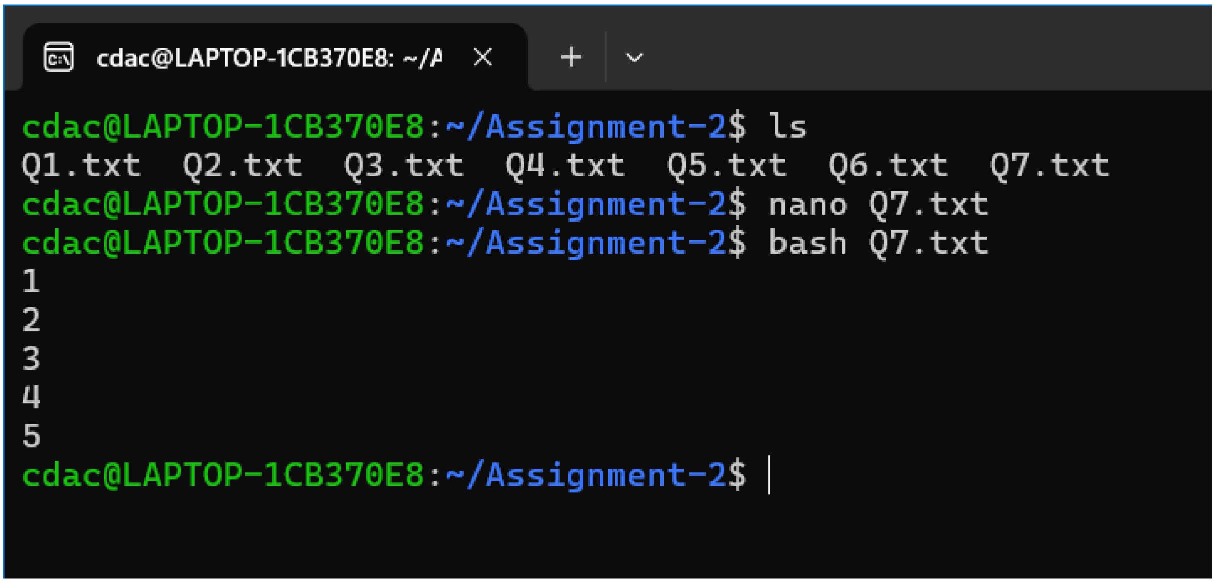


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q2.txt  Q3.txt  Q4.txt  Q5.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q6.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q6.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q6.txt
1
2
3
4
5
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

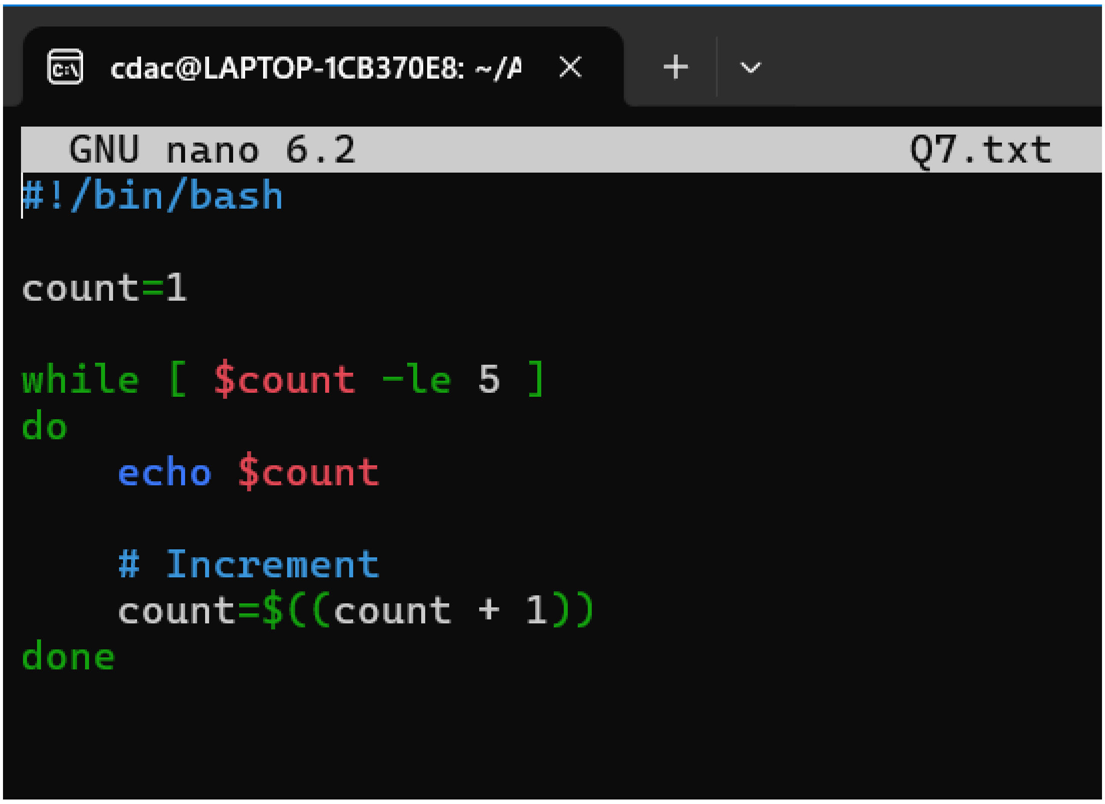


```
  GNU nano 6.2                                Q6.txt
#!/bin/bash
a=0
for a in 1 2 3 4 5
do
        echo $a
done
```

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

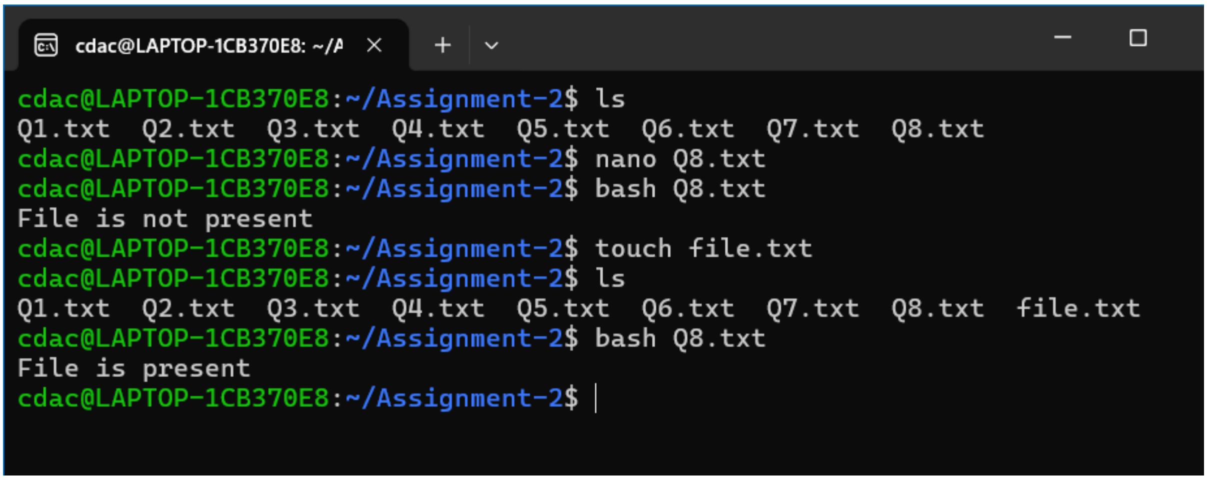


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q2.txt  Q3.txt  Q4.txt  Q5.txt  Q6.txt  Q7.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q7.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q7.txt
1
2
3
4
5
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

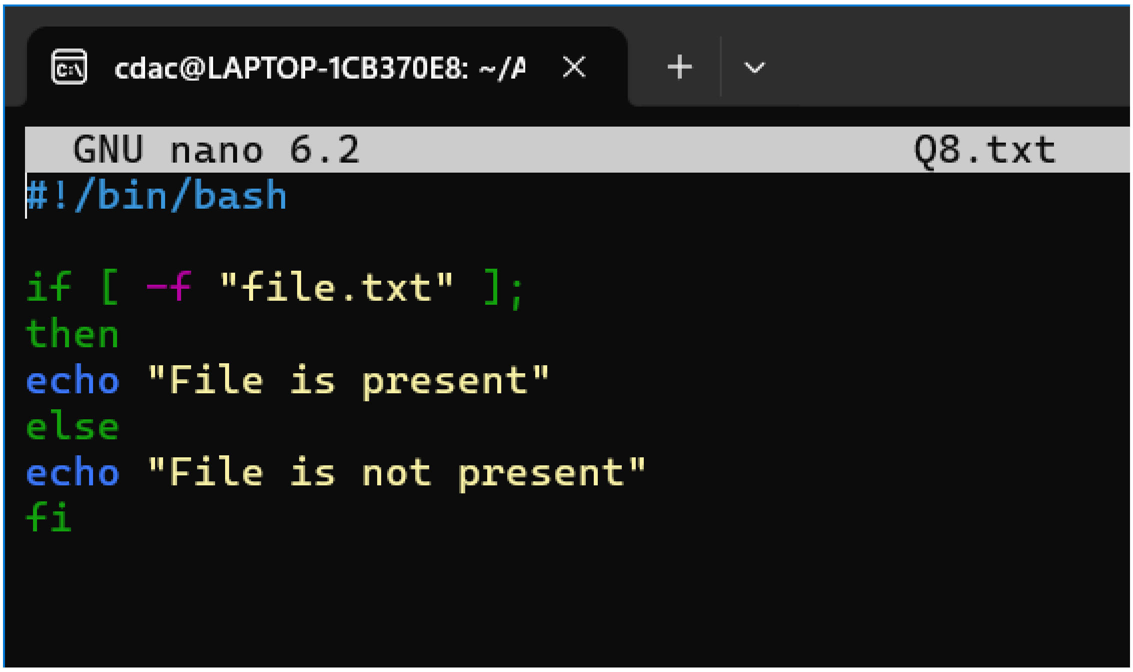


```
  GNU nano 6.2                                Q7.txt
#!/bin/bash

count=1

while [ $count -le 5 ]
do
    echo $count

    # Increment
    count=$((count + 1))
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".

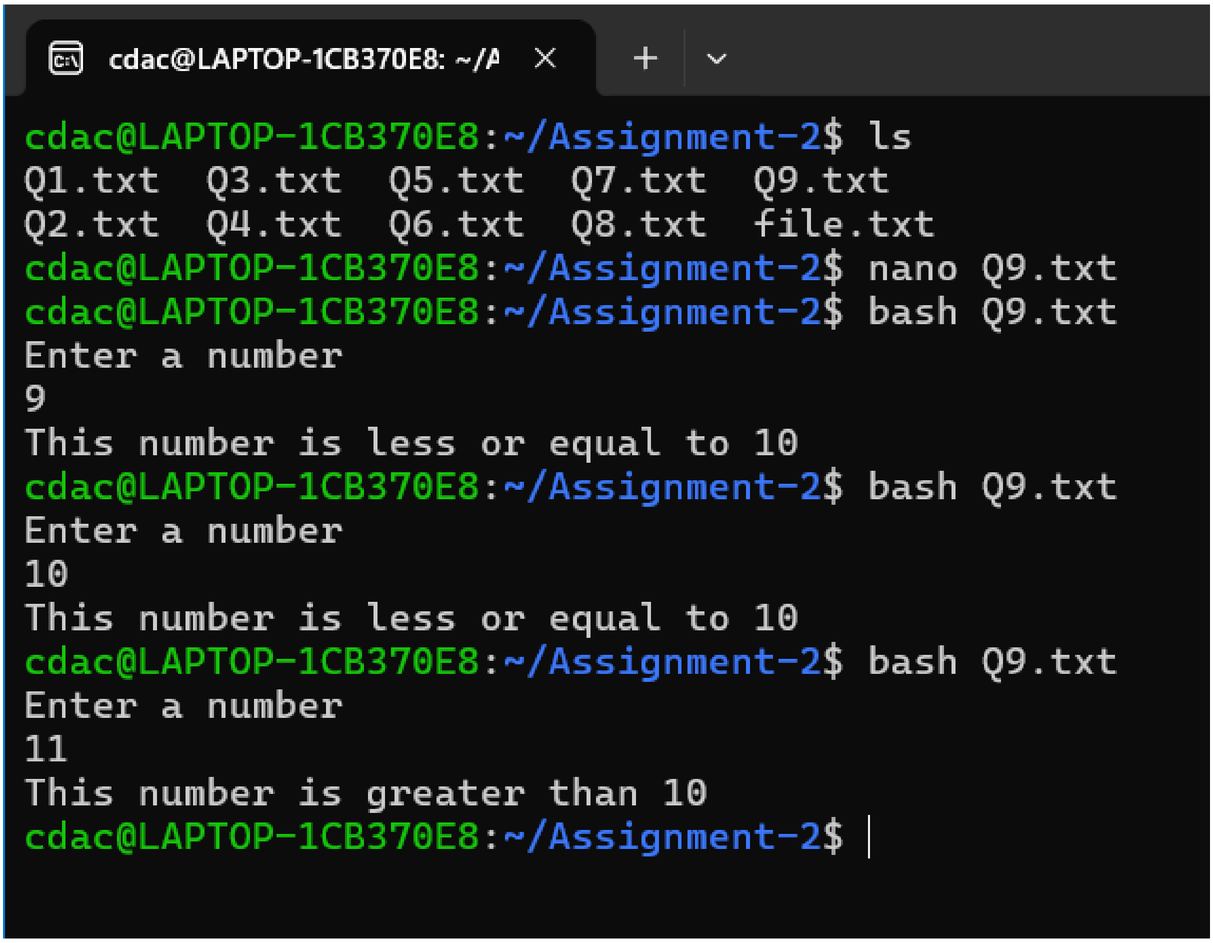


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q2.txt  Q3.txt  Q4.txt  Q5.txt  Q6.txt  Q7.txt  Q8.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q8.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q8.txt
File is not present
cdac@LAPTOP-1CB370E8:~/Assignment-2$ touch file.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q2.txt  Q3.txt  Q4.txt  Q5.txt  Q6.txt  Q7.txt  Q8.txt  file.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q8.txt
File is present
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

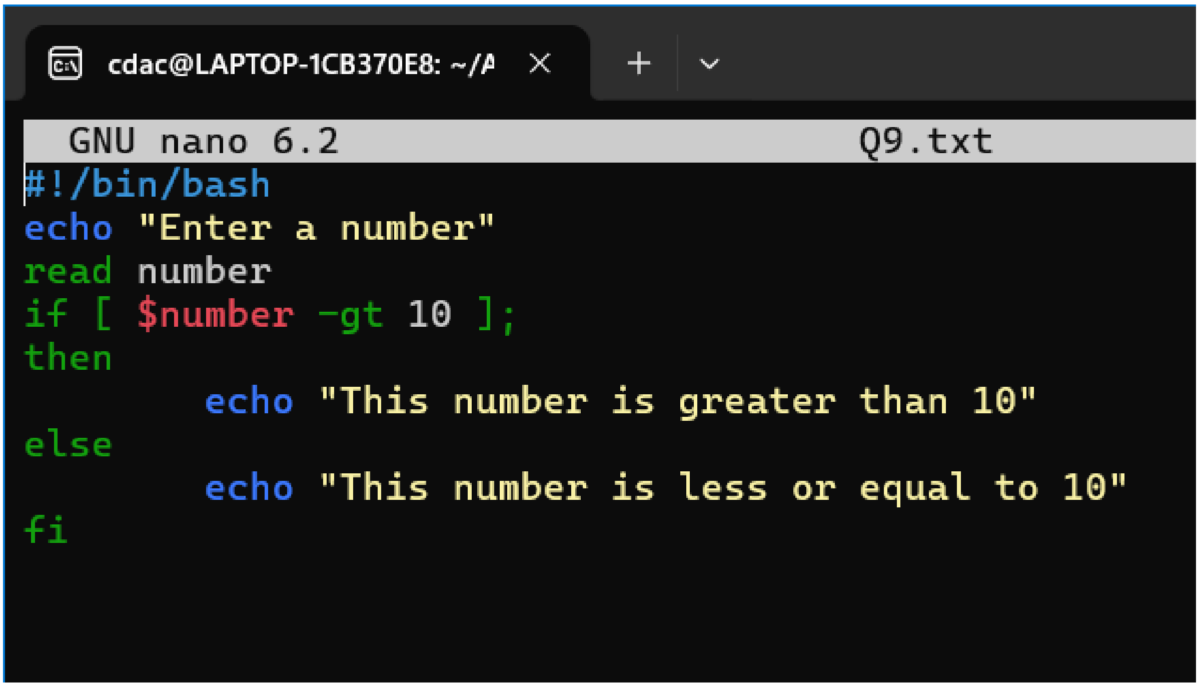


```
  GNU nano 6.2                                   Q8.txt
#!/bin/bash

if [ -f "file.txt" ];
then
echo "File is present"
else
echo "File is not present"
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.

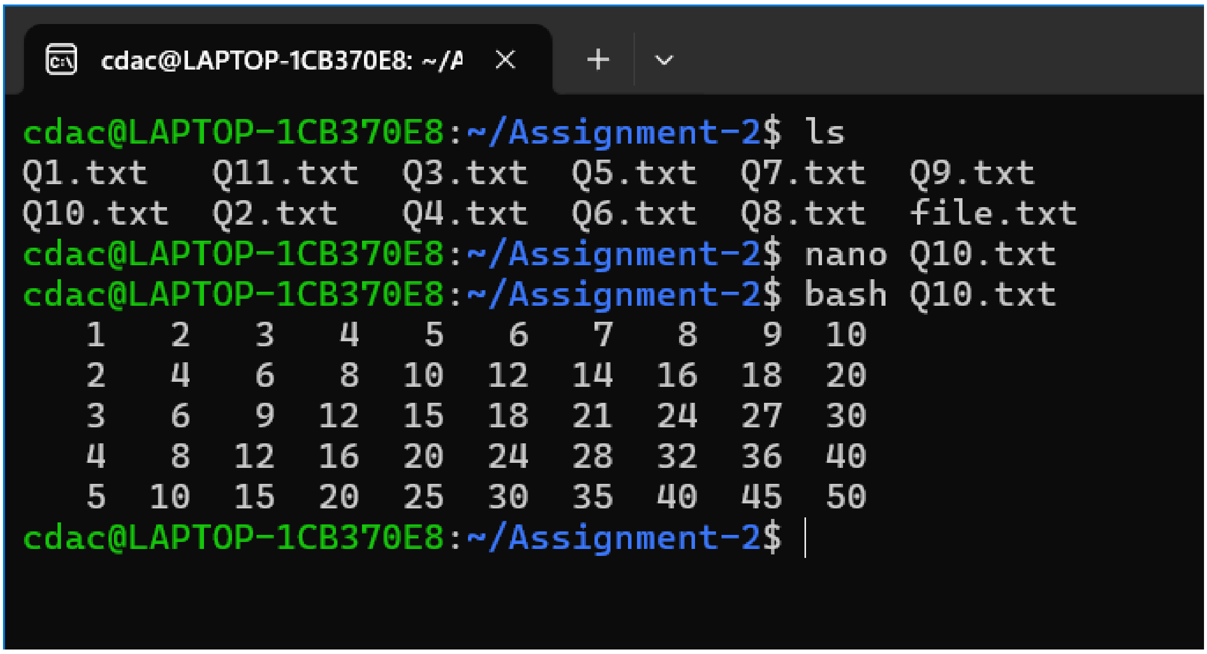


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q3.txt  Q5.txt  Q7.txt  Q9.txt
Q2.txt  Q4.txt  Q6.txt  Q8.txt  file.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q9.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q9.txt
Enter a number
9
This number is less or equal to 10
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q9.txt
Enter a number
10
This number is less or equal to 10
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q9.txt
Enter a number
11
This number is greater than 10
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

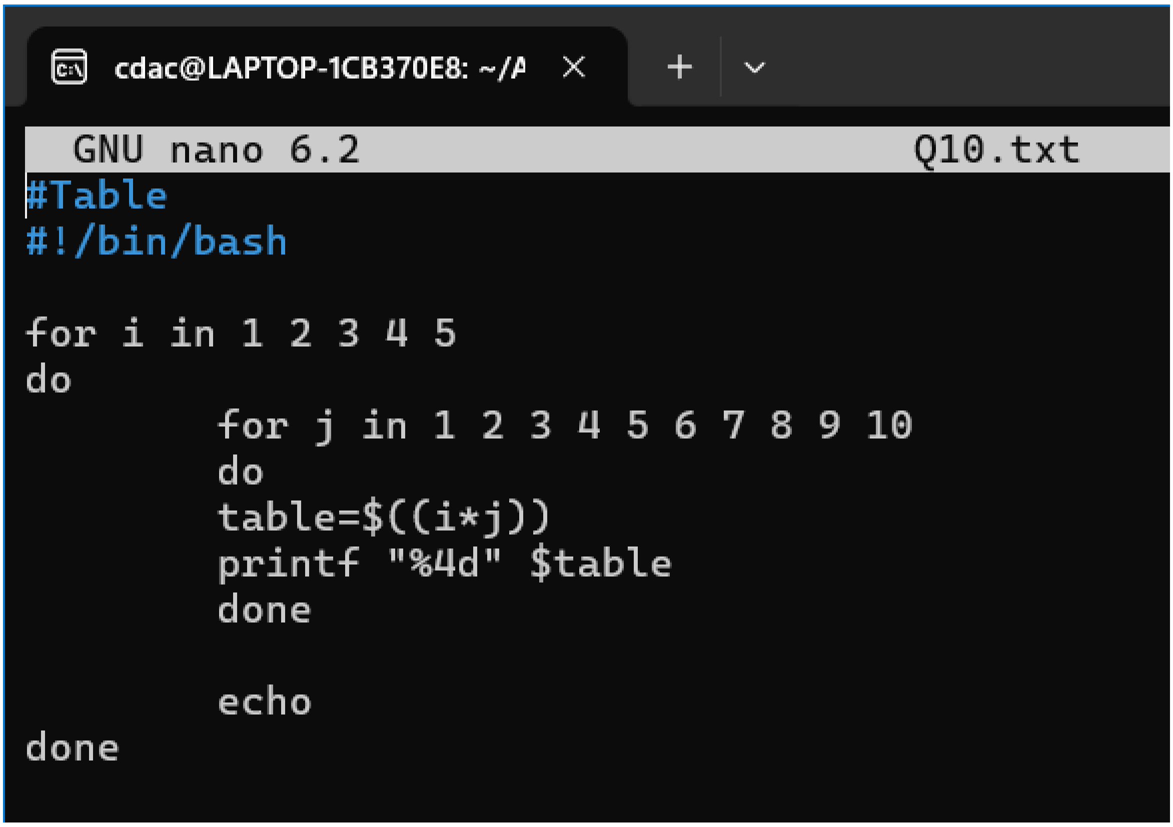


```
  GNU nano 6.2                                Q9.txt
#!/bin/bash
echo "Enter a number"
read number
if [ $number -gt 10 ];
then
        echo "This number is greater than 10"
else
        echo "This number is less or equal to 10"
fi
```

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.

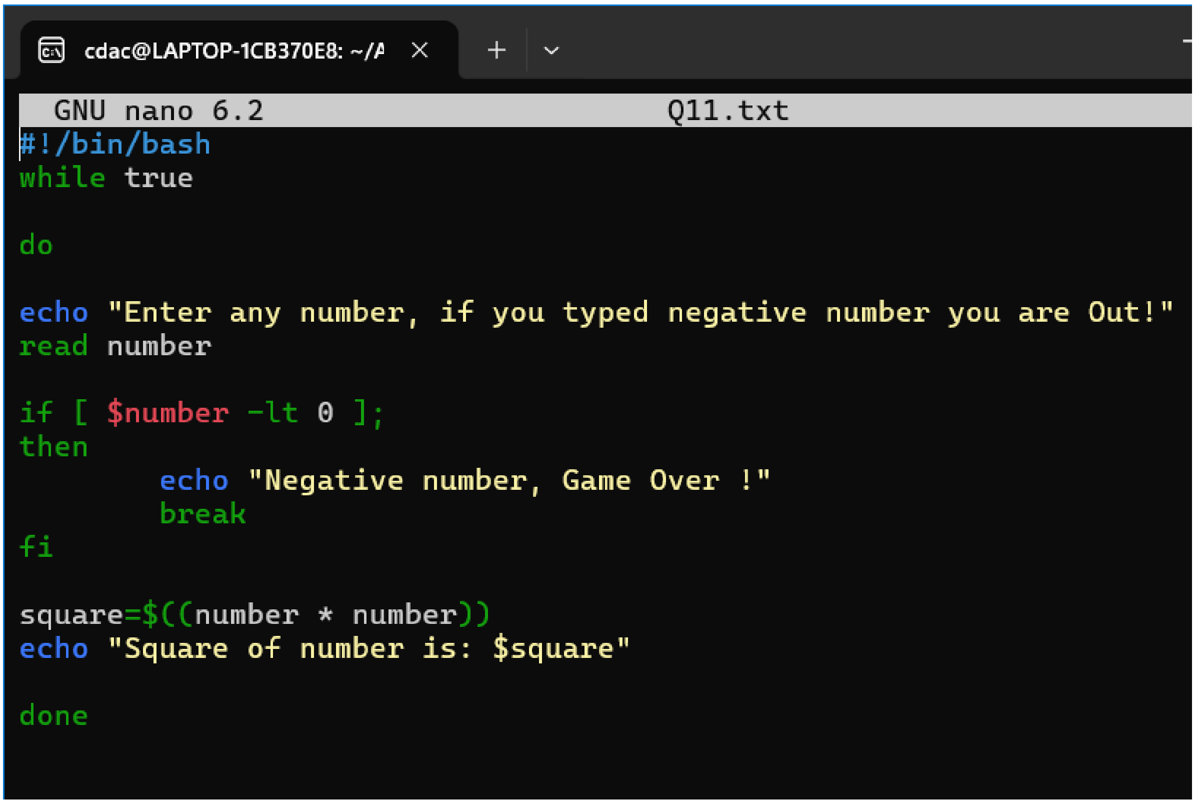


```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt    Q11.txt   Q3.txt   Q5.txt   Q7.txt   Q9.txt
Q10.txt   Q2.txt    Q4.txt   Q6.txt   Q8.txt   file.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q10.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q10.txt
   1   2   3   4   5   6   7   8   9  10
   2   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
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

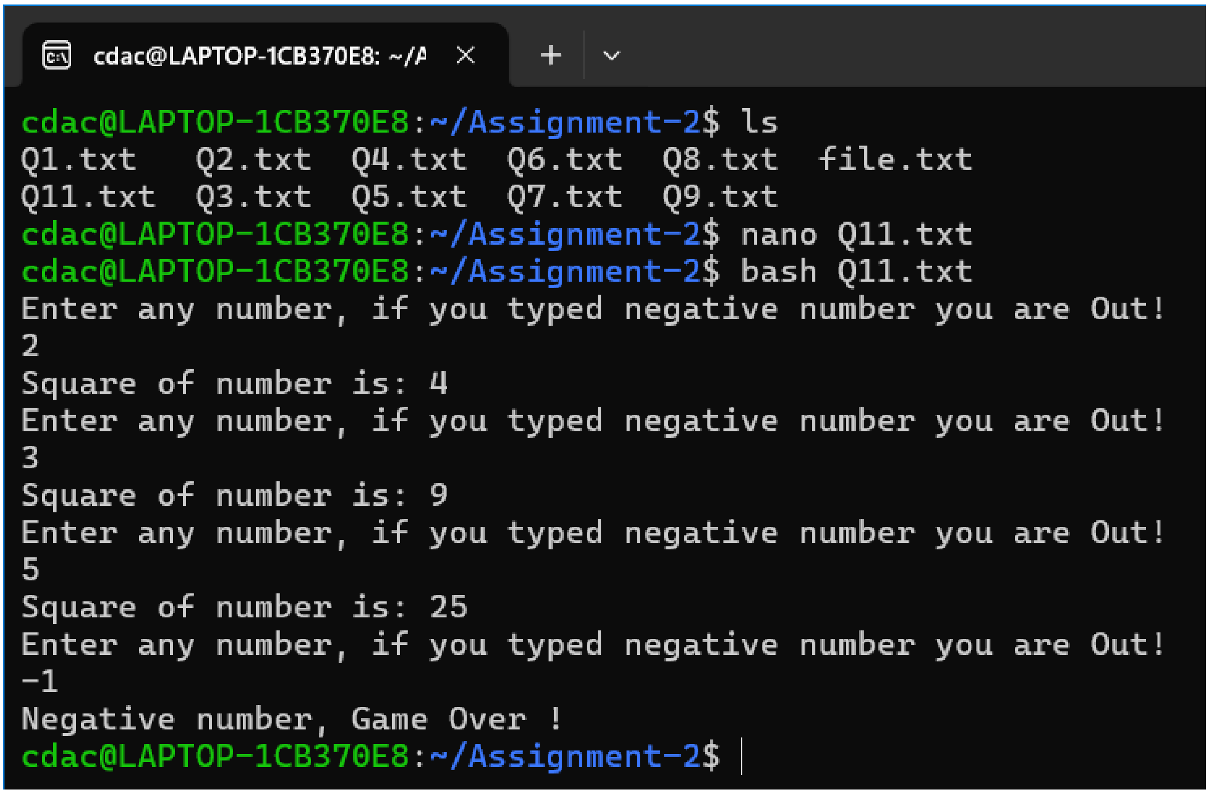


```
  GNU nano 6.2                          Q10.txt
#Table
#!/bin/bash

for i in 1 2 3 4 5
do
        for j in 1 2 3 4 5 6 7 8 9 10
        do
        table=$((i*j))
        printf "%4d" $table
        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.



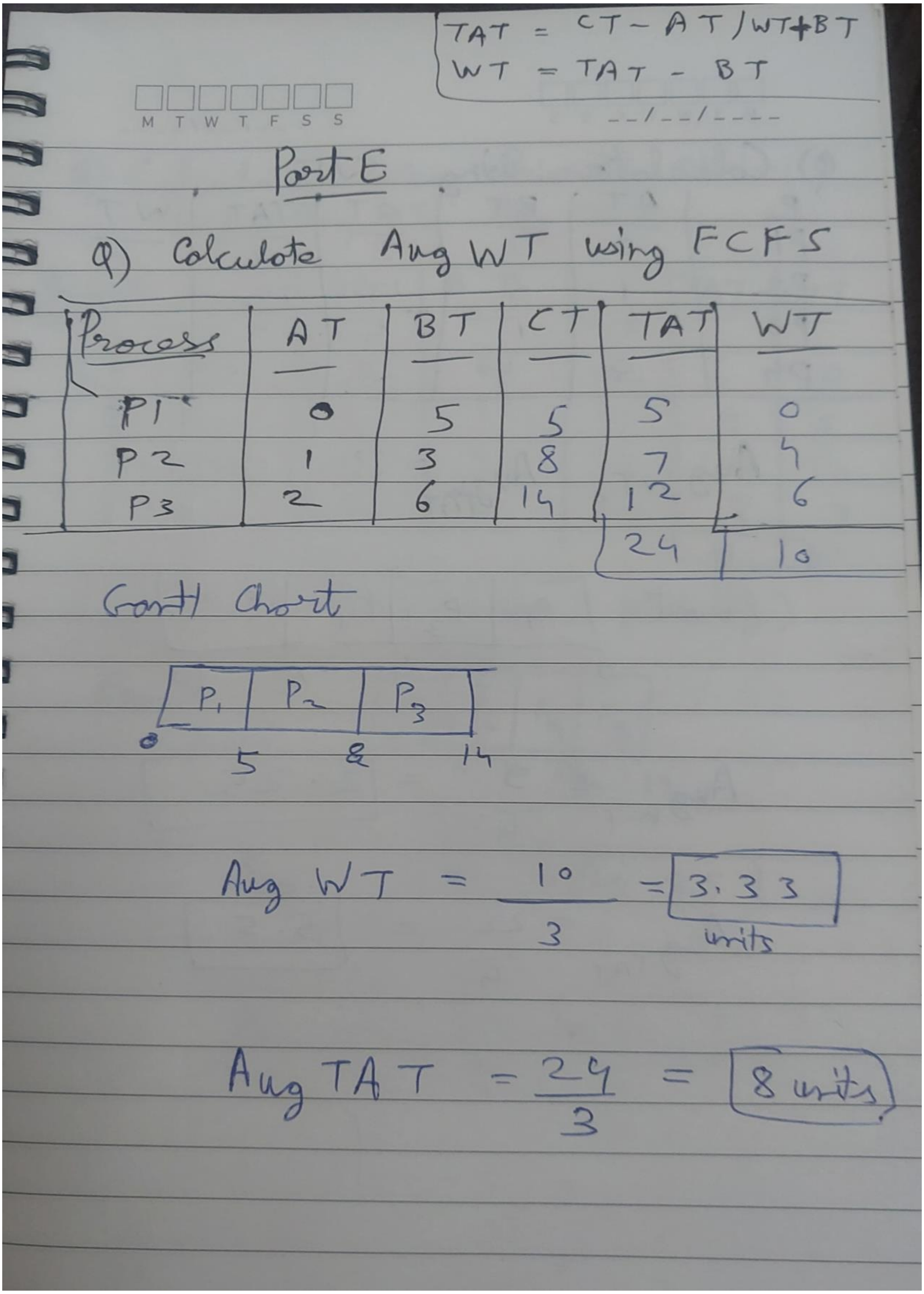
```
  GNU nano 6.2                              Q11.txt
#!/bin/bash
while true

do

echo "Enter any number, if you typed negative number you are Out!"
read number

if [ $number -lt 0 ];
then
        echo "Negative number, Game Over !"
        break
fi

square=$((number * number))
echo "Square of number is: $square"

done
```



```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt   Q2.txt  Q4.txt  Q6.txt  Q8.txt  file.txt
Q11.txt  Q3.txt  Q5.txt  Q7.txt  Q9.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano Q11.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash Q11.txt
Enter any number, if you typed negative number you are Out!
2
Square of number is: 4
Enter any number, if you typed negative number you are Out!
3
Square of number is: 9
Enter any number, if you typed negative number you are Out!
5
Square of number is: 25
Enter any number, if you typed negative number you are Out!
-1
Negative number, Game Over !
cdac@LAPTOP-1CB370E8:~/Assignment-2$
```

Part E

Q.1



TAT = CT − AT / WT+BT

WT = TAT − BT

□□□□□□□
M T W T F S S

--/--/----

Part E

Q) Calculate Avg WT using FCFS

Process	AT	BT	CT	TAT	WT
P1	0	5	5	5	0
P2	1	3	8	7	4
P3	2	6	14	12	6
				24	10

Gantt Chart

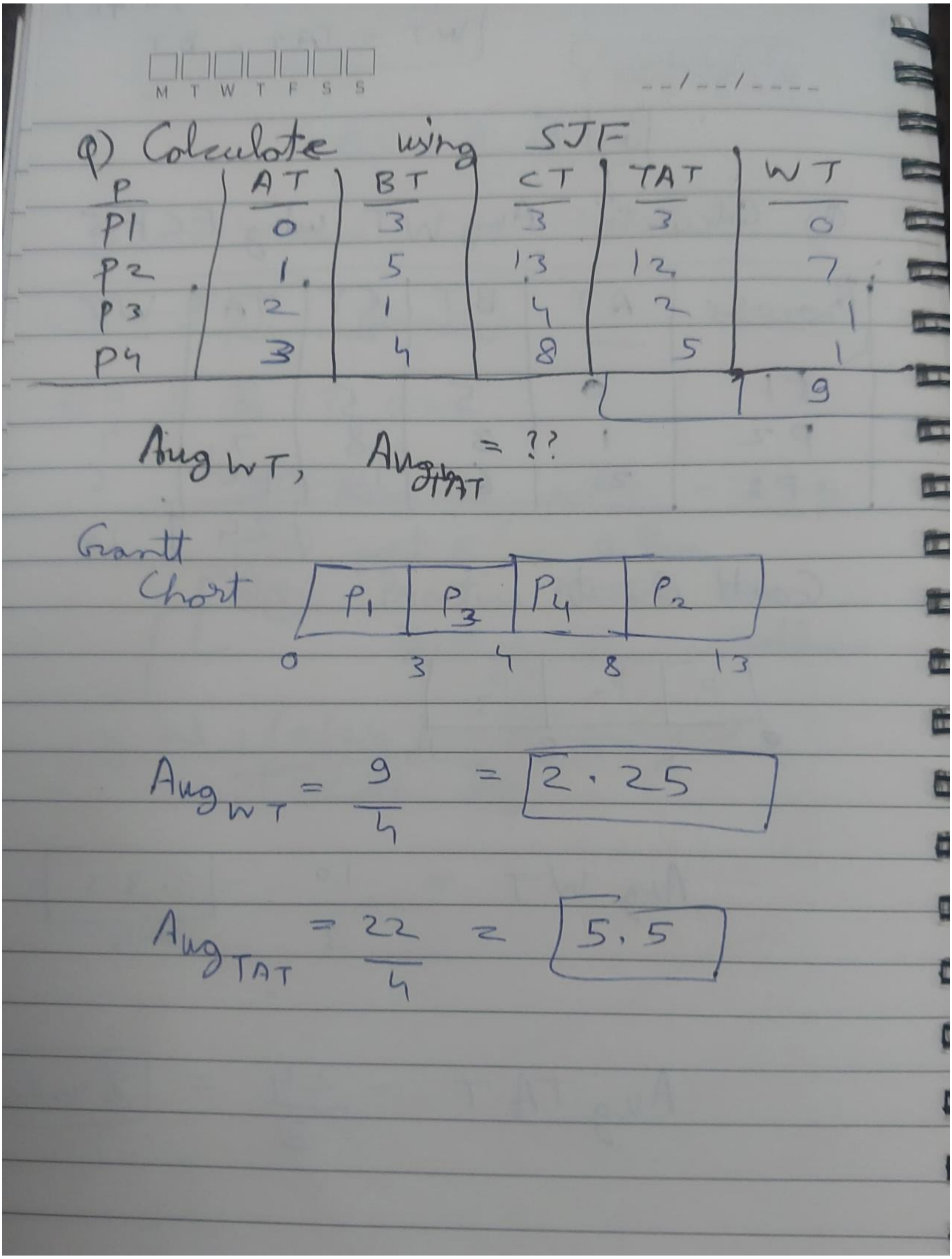
P_1	P_2	P_3

0 5 8 14

$$\text{Avg WT} = \frac{10}{3} = \boxed{3.33} \text{ units}$$

$$\text{Avg TAT} = \frac{24}{3} = \boxed{8 \text{ units}}$$

Q.2



□□□□□□□
M T W T F S S

--/--/----

Q) Calculate using SJF

P	AT	BT	CT	TAT	WT
P1	0	3	3	3	0
P2	1	5	13	12	7
P3	2	1	4	2	1
P4	3	4	8	5	1
					9

Avg_{WT}, Avg_{TAT} = ??

Gantt Chart

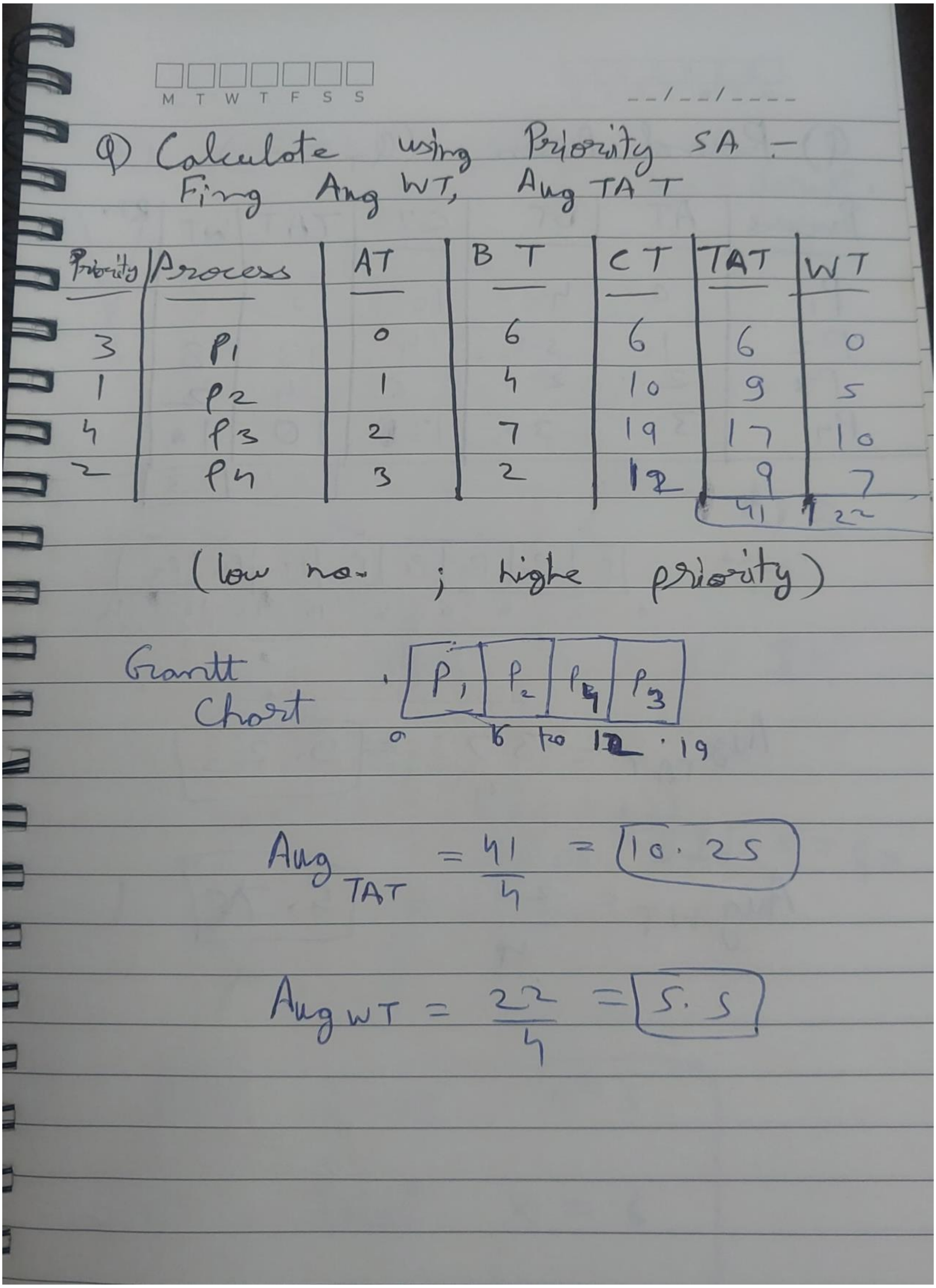
P1	P3	P4	P2

0 3 4 8 13

$$Avg_{WT} = \frac{9}{4} = \boxed{2.25}$$

$$Avg_{TAT} = \frac{22}{4} = \boxed{5.5}$$

Q.3



□□□□□□□
M T W T F S S

--/--/----

Q) Calculate using Priority SA :-
Fing Avg WT, Avg TAT

Priority	Process	AT	BT	CT	TAT	WT
3	P1	0	6	6	6	0
1	P2	1	4	10	9	5
4	P3	2	7	19	17	10
2	P4	3	2	12	9	7
					41	22

(low no. ; highe priority)

Gantt Chart

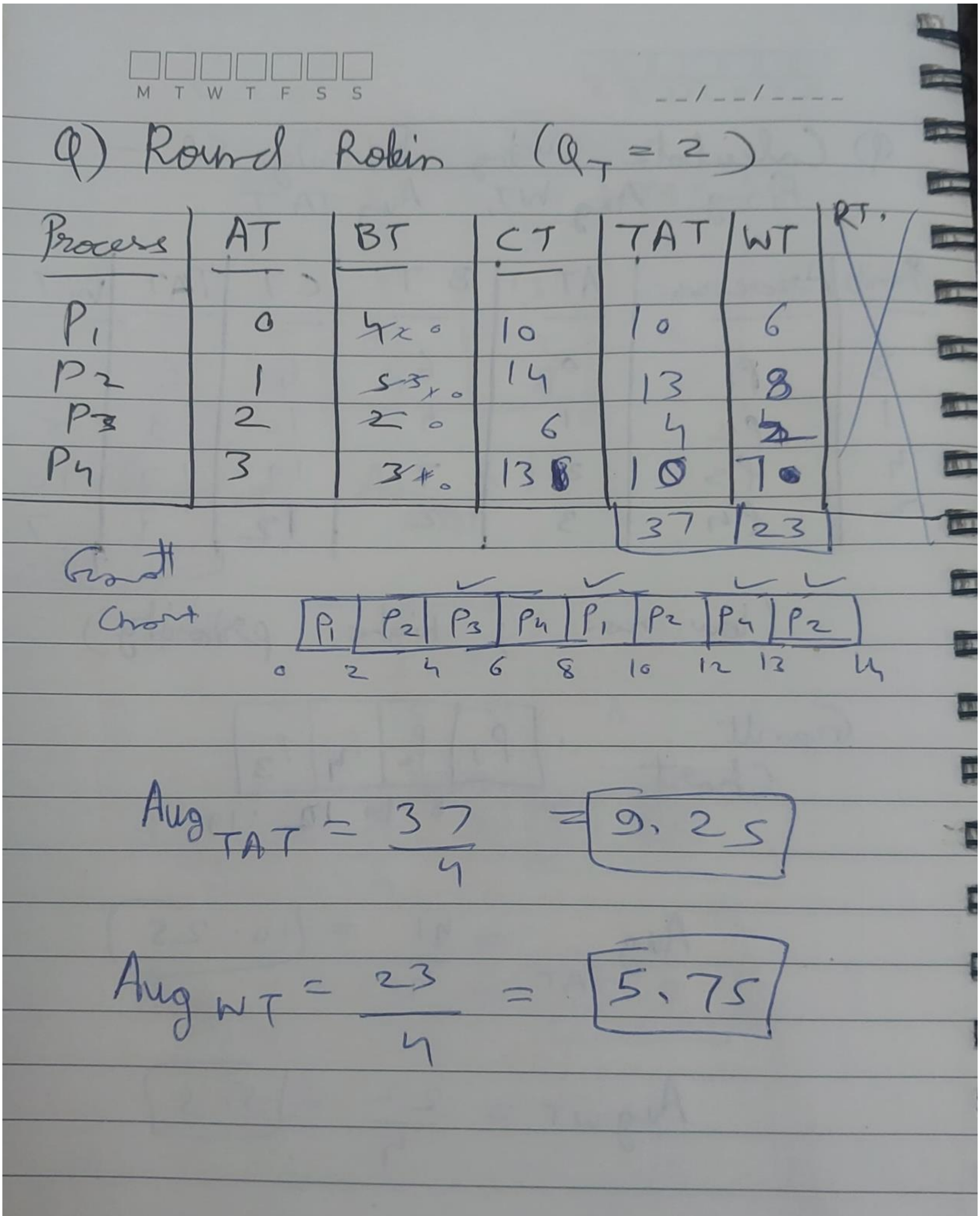
P1	P2	P4	P3

0 6 10 12 19

$$\text{Avg}_{TAT} = \frac{41}{4} = \boxed{10.25}$$

$$\text{Avg}_{WT} = \frac{22}{4} = \boxed{5.5}$$

Q.4



Q) Round Robin ($Q_T = 2$)

Process	AT	BT	CT	TAT	WT	~~RT.~~
P_1	0	~~4~~ ~~2~~ 0	10	10	6	
P_2	1	~~5~~ ~~3~~ ~~1~~ 0	14	13	8	
P_3	2	~~2~~ 0	6	4	2	
P_4	3	~~3~~ ~~1~~ 0	13	10	7	
				37	23	

Gantt Chart

P_1	P_2	P_3	P_4	P_1	P_2	P_4	P_2

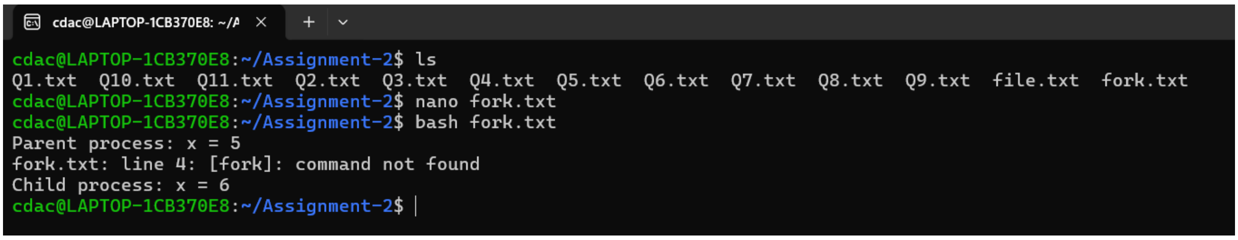
0 2 4 6 8 10 12 13 14

$$Avg_{TAT} = \frac{37}{4} = \boxed{9.25}$$

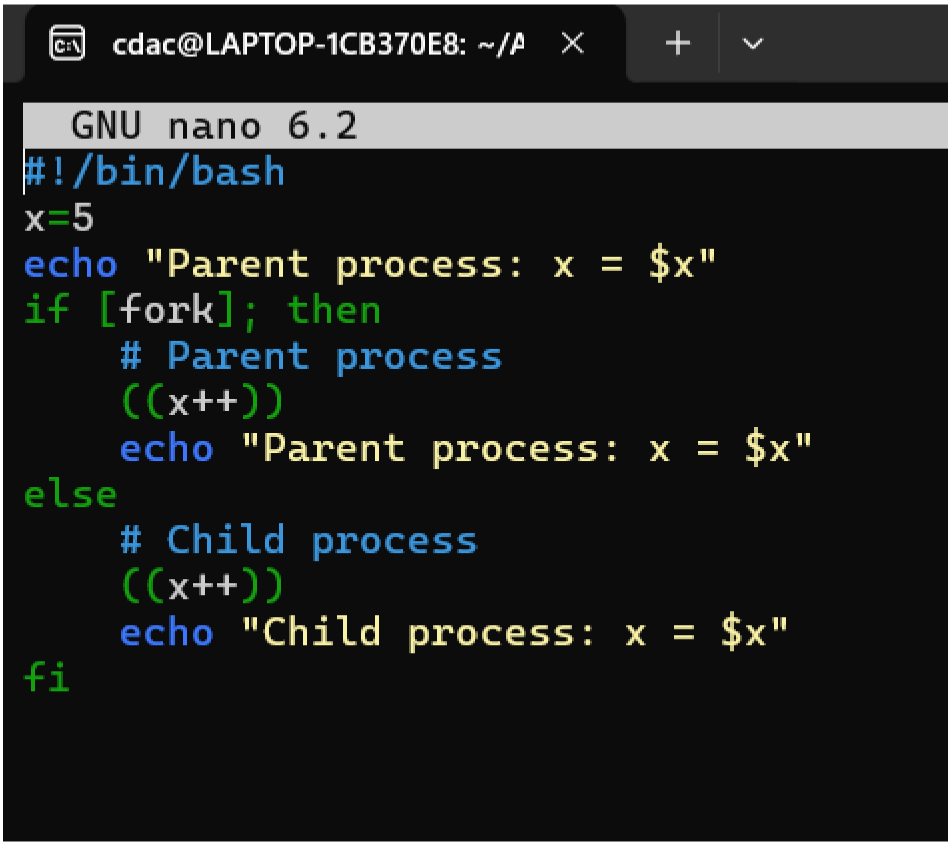
$$Avg_{WT} = \frac{23}{4} = \boxed{5.75}$$

Q.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?



```
cdac@LAPTOP-1CB370E8:~/Assignment-2$ ls
Q1.txt  Q10.txt  Q11.txt  Q2.txt  Q3.txt  Q4.txt  Q5.txt  Q6.txt  Q7.txt  Q8.txt  Q9.txt  file.txt  fork.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ nano fork.txt
cdac@LAPTOP-1CB370E8:~/Assignment-2$ bash fork.txt
Parent process: x = 5
fork.txt: line 4: [fork]: command not found
Child process: x = 6
cdac@LAPTOP-1CB370E8:~/Assignment-2$ 
```



```
  GNU nano 6.2
#!/bin/bash
x=5
echo "Parent process: x = $x"
if [fork]; then
    # Parent process
    ((x++))
    echo "Parent process: x = $x"
else
    # Child process
    ((x++))
    echo "Child process: x = $x"
fi
```