Part A.

What will the following commands do?

- echo "Hello, World!"

Ans: prints Hello, World! on the terminal

- name="Productive"

Ans: defines a variable name with the value Productive

- touch file.txt

Ans: creates a file named as file.txt

- ls -a

Ans: lists all the files in the current directory, including hidden files

- rm file.txt

Ans: deletes the file named as file.txt

- cp file1.txt file2.txt

Ans: creates a copy of file1.txt with a name file2.txt

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

Ans: moves the file file.txt to the specified directory

- chmod 755 script.sh

Ans: changes the file permission, enables owner to read, write, execute; group to read, execute; others to read, execute.

- grep "pattern" file.txt

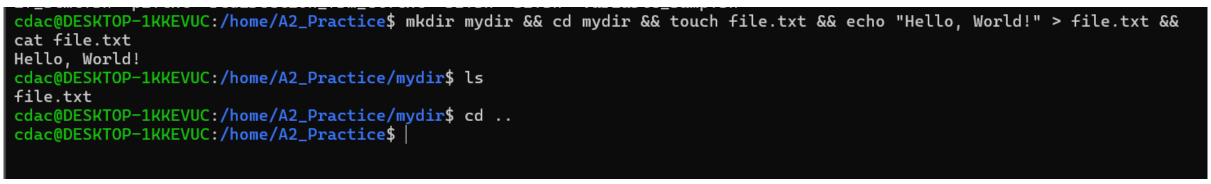
Ans: searches for the word pattern in the file file.txt

- kill PID

Ans: terminates a process with the given PID

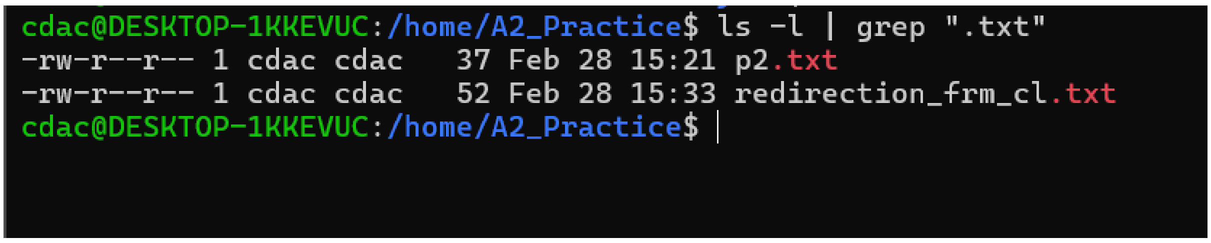
- mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt

Ans: makes a directory called mydir and goes into it creates a file named file.txt and copies the line echo "Hello, World!" in the file file.txt and prints the contents of the file.



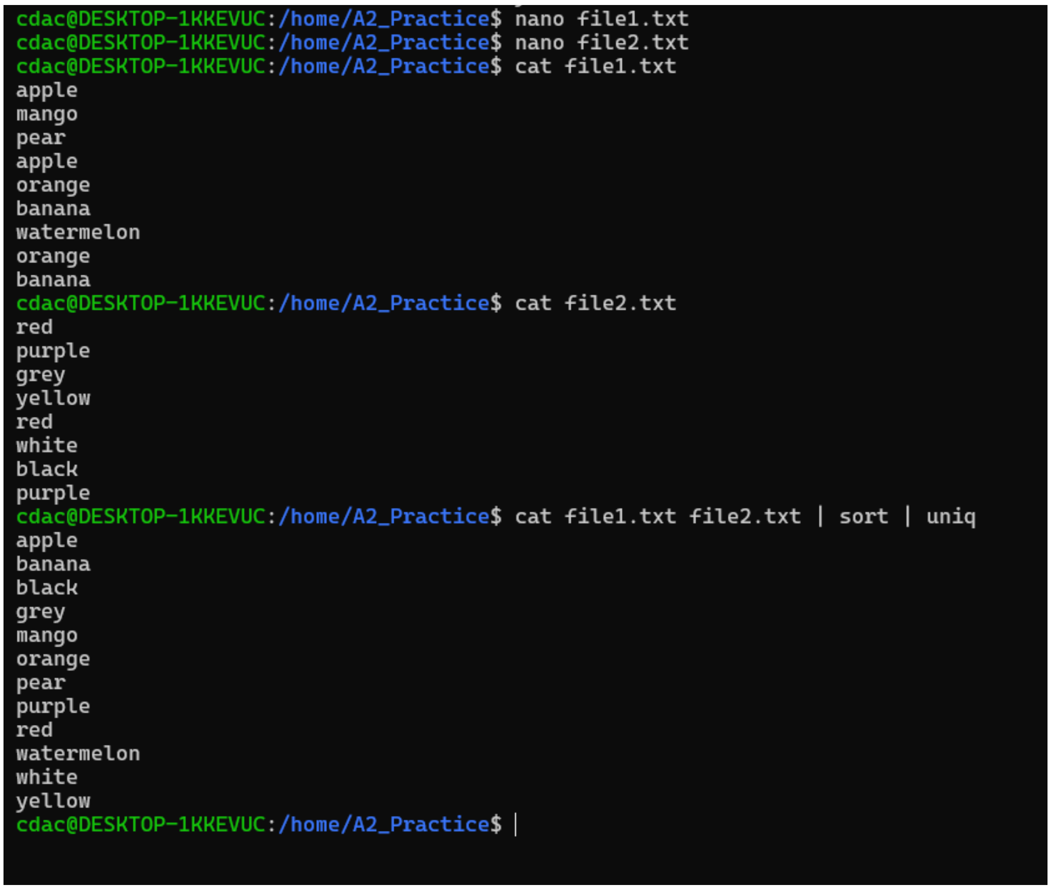
```
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt &&
cat file.txt
Hello, World!
cdac@DESKTOP-1KKEVUC:/home/A2_Practice/mydir$ ls
file.txt
cdac@DESKTOP-1KKEVUC:/home/A2_Practice/mydir$ cd ..
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ |
```

• ls -l | grep ".txt"
Ans: in the listed files and folders, .txt files are enlisted



```
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ ls -l | grep ".txt"
-rw-r--r-- 1 cdac cdac   37 Feb 28 15:21 p2.txt
-rw-r--r-- 1 cdac cdac   52 Feb 28 15:33 redirection_frm_cl.txt
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$
```

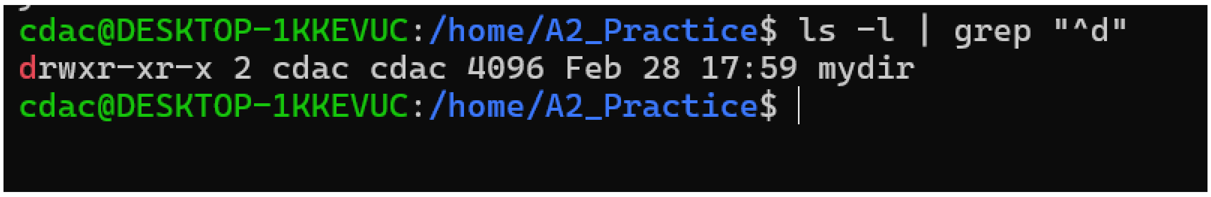
• cat file1.txt file2.txt | sort | uniq
Ans: from the files file1.txt and file2.txt it sorts the file and outputs only the unique lines



```
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ nano file1.txt
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ nano file2.txt
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ cat file1.txt
apple
mango
pear
apple
orange
banana
watermelon
orange
banana
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ cat file2.txt
red
purple
grey
yellow
red
white
black
purple
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ cat file1.txt file2.txt | sort | uniq
apple
banana
black
grey
mango
orange
pear
purple
red
watermelon
white
yellow
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$
```

• ls -l | grep "^d"

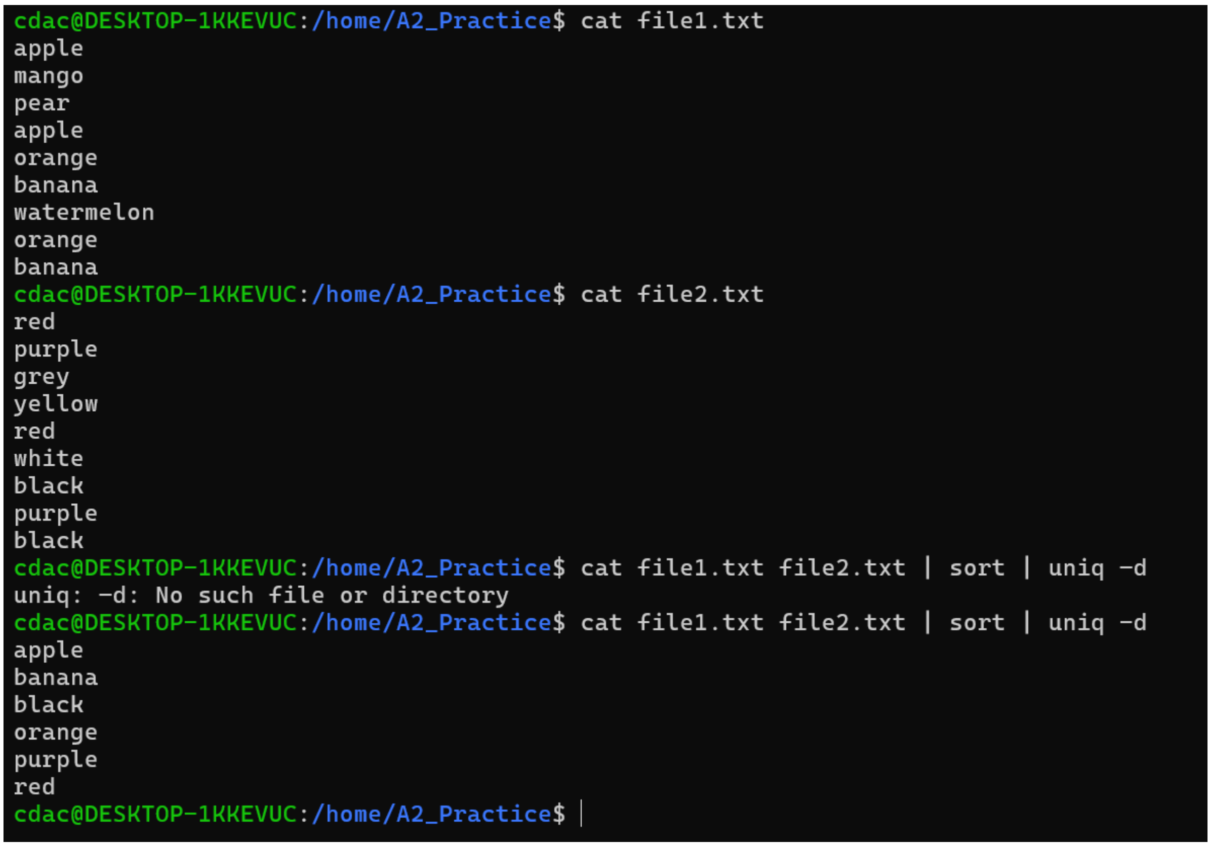
Ans: out of the files and directories present in the current directory, it outputs the directories (prints the items in the current directory that are directories)



```
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ ls -l | grep "^d"
drwxr-xr-x 2 cdac cdac 4096 Feb 28 17:59 mydir
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$
```

• grep -r "pattern" /path/to/directory/
Ans: For each directory operand, read and process all files in that directory, recursively. Note that if no file operand is given, grep searches the working directory.

• cat file1.txt file2.txt | sort | uniq –d
Ans: It outputs the duplicate lines



```
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ cat file1.txt
apple
mango
pear
apple
orange
banana
watermelon
orange
banana
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ cat file2.txt
red
purple
grey
yellow
red
white
black
purple
black
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ cat file1.txt file2.txt | sort | uniq –d
uniq: –d: No such file or directory
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$ cat file1.txt file2.txt | sort | uniq -d
apple
banana
black
orange
purple
red
cdac@DESKTOP-1KKEVUC:/home/A2_Practice$
```

• chmod 644 file.txt
Ans: Changes the permissions of the file for owner to read, write; group to read; others to read

• cp -r source_directory destination_directory
Ans: copies recursively all the files/subdirectories in the current directory to the destination directory (The -r option tells cp to descend into subdirectories and copy everything within the source_directory)

• find /path/to/search -name "*.txt"
Ans: it searches for all files ending with ".txt" within the specified directory and its subdirectories.

• chmod u+x file.txt
Ans: adds the file permissions of the current user (owner) to execute

• echo $PATH
Ans: displays the value of the PATH environment variable

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 to copy files and directories)

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 is used

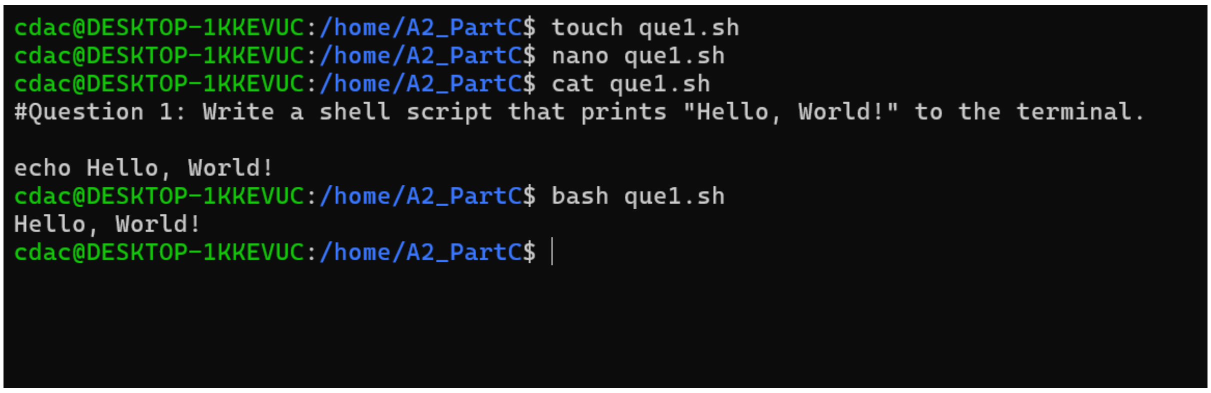
2. **cpy** is used to copy files and directories.
=> cp is used

3. **mkfile** is used to create a new file.
=> touch is used

4. **catx** is used to concatenate files.
=> cat is used

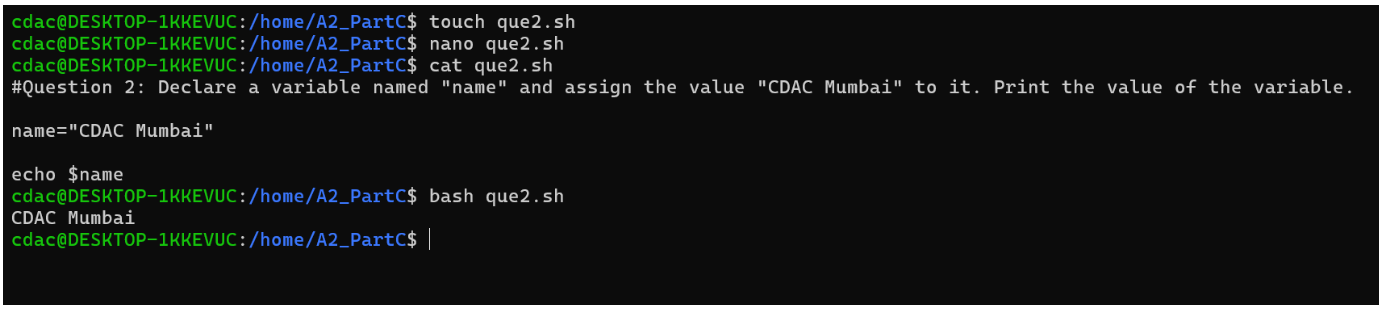
5. **rn** is used to rename files.
=> mv is used

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



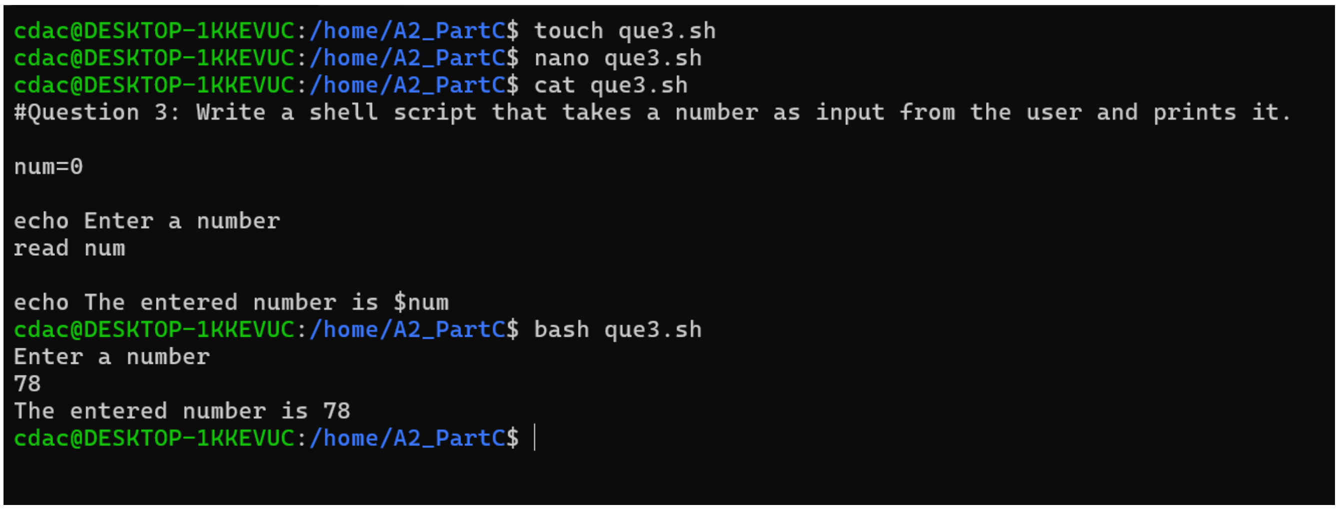
```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ touch que1.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que1.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que1.sh
#Question 1: Write a shell script that prints "Hello, World!" to the terminal.

echo Hello, World!
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que1.sh
Hello, World!
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$
```

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



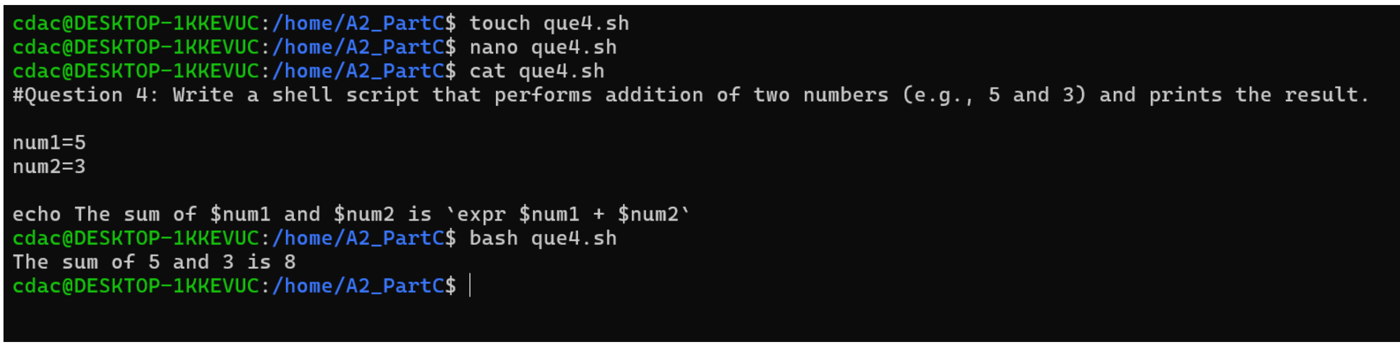
```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ touch que2.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que2.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que2.sh
#Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

name="CDAC Mumbai"

echo $name
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que2.sh
CDAC Mumbai
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$
```

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



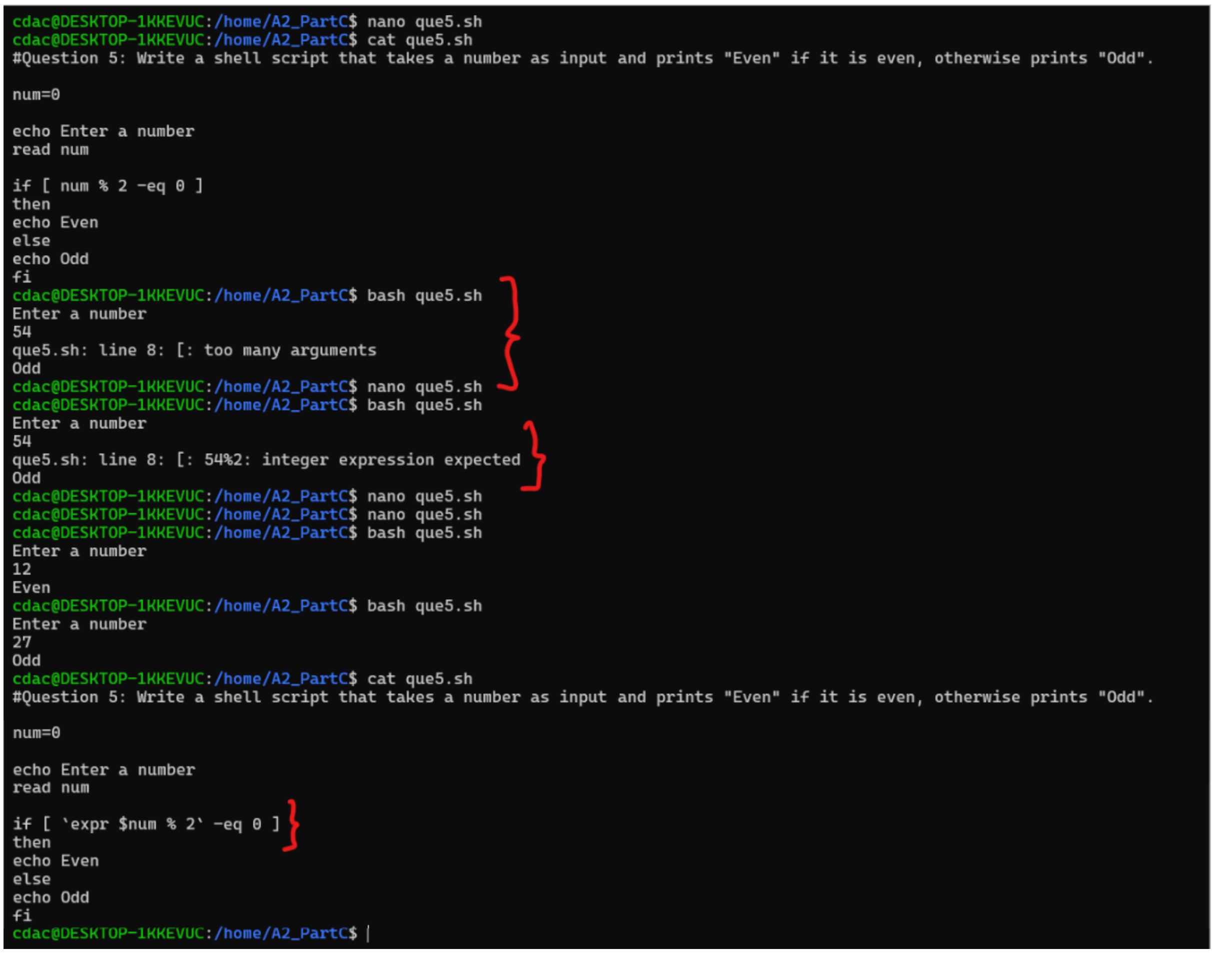
```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ touch que3.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que3.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que3.sh
#Question 3: Write a shell script that takes a number as input from the user and prints it.

num=0

echo Enter a number
read num

echo The entered number is $num
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que3.sh
Enter a number
78
The entered number is 78
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$
```

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



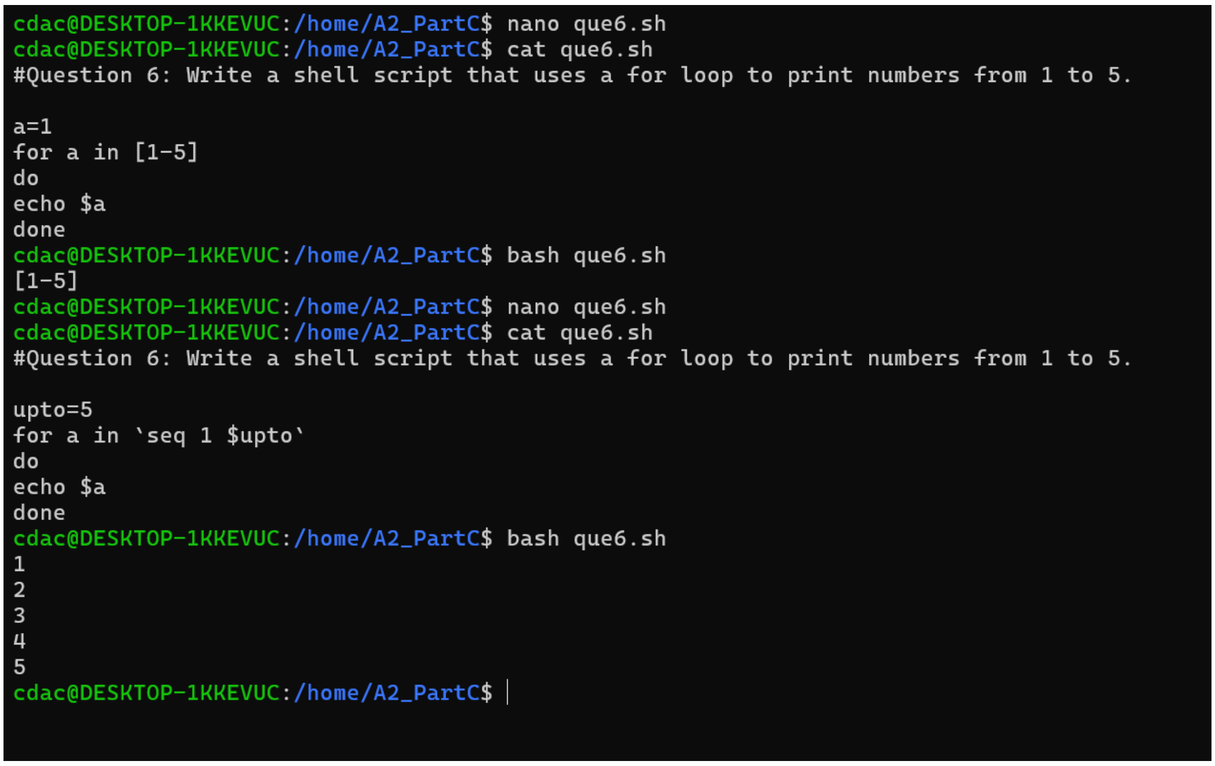
```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ touch que4.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que4.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que4.sh
#Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.

num1=5
num2=3

echo The sum of $num1 and $num2 is `expr $num1 + $num2`
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que4.sh
The sum of 5 and 3 is 8
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ |
```

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



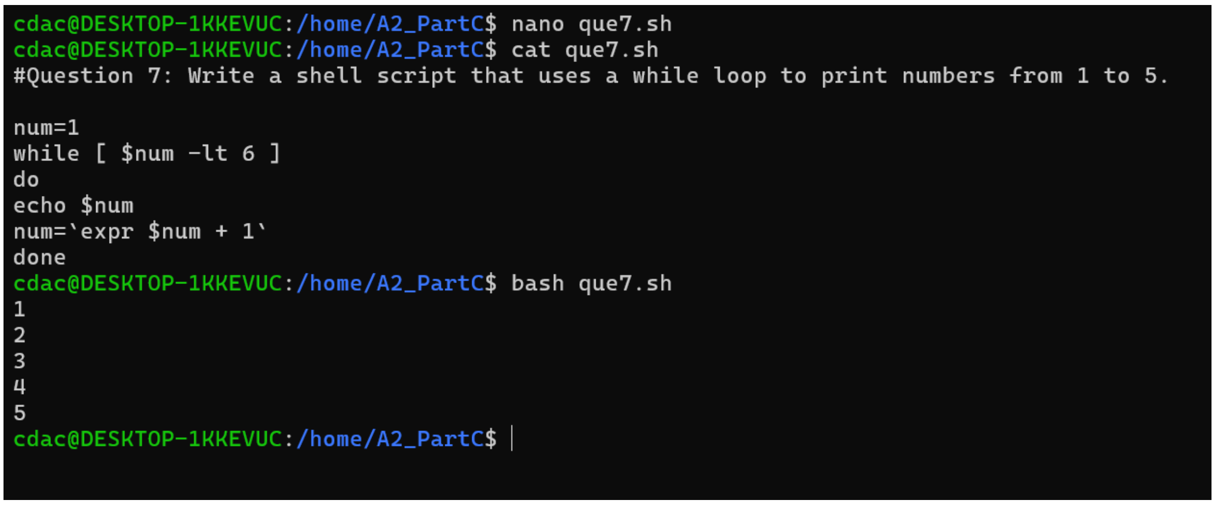
```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que5.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que5.sh
#Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

num=0

echo Enter a number
read num

if [ num % 2 -eq 0 ]
then
echo Even
else
echo Odd
fi
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que5.sh
Enter a number
54
que5.sh: line 8: [: too many arguments
Odd
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que5.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que5.sh
Enter a number
54
que5.sh: line 8: [: 54%2: integer expression expected
Odd
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que5.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que5.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que5.sh
Enter a number
12
Even
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que5.sh
Enter a number
27
Odd
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que5.sh
#Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

num=0

echo Enter a number
read num

if [ `expr $num % 2` -eq 0 ]
then
echo Even
else
echo Odd
fi
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ |
```

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



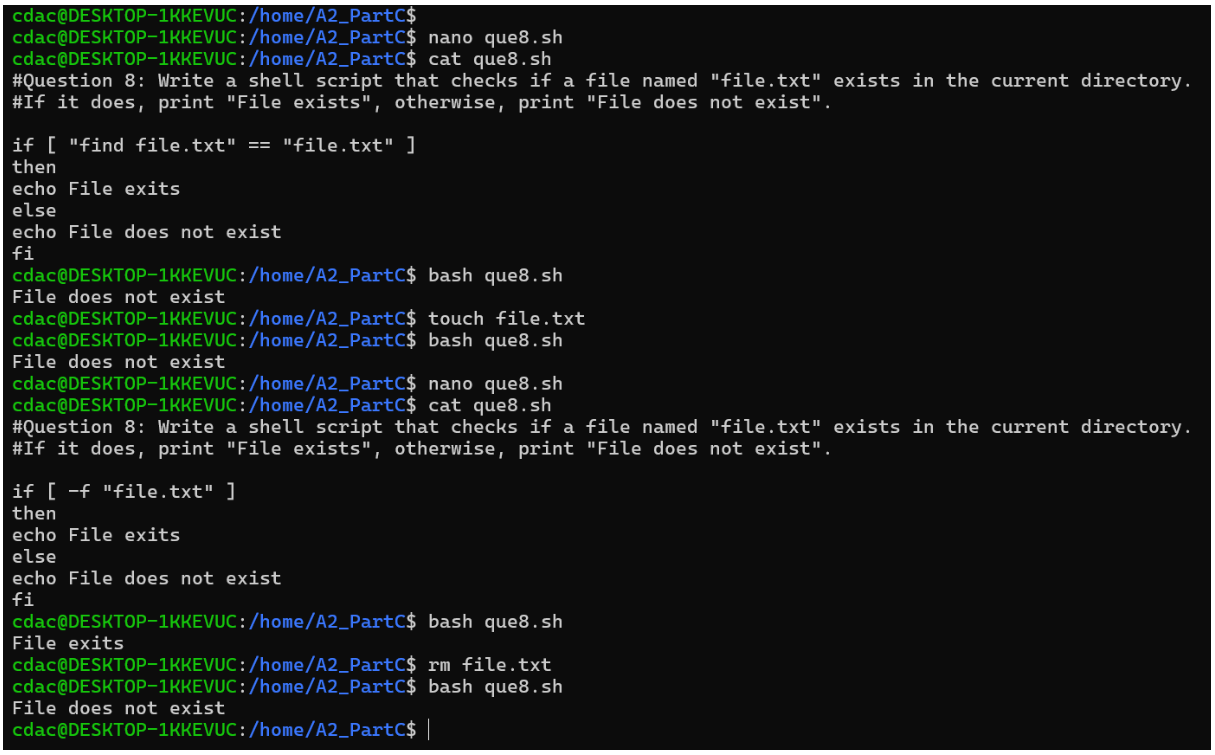
```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que6.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que6.sh
#Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

a=1
for a in [1-5]
do
echo $a
done
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que6.sh
[1-5]
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que6.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que6.sh
#Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

upto=5
for a in `seq 1 $upto`
do
echo $a
done
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que6.sh
1
2
3
4
5
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$
```

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



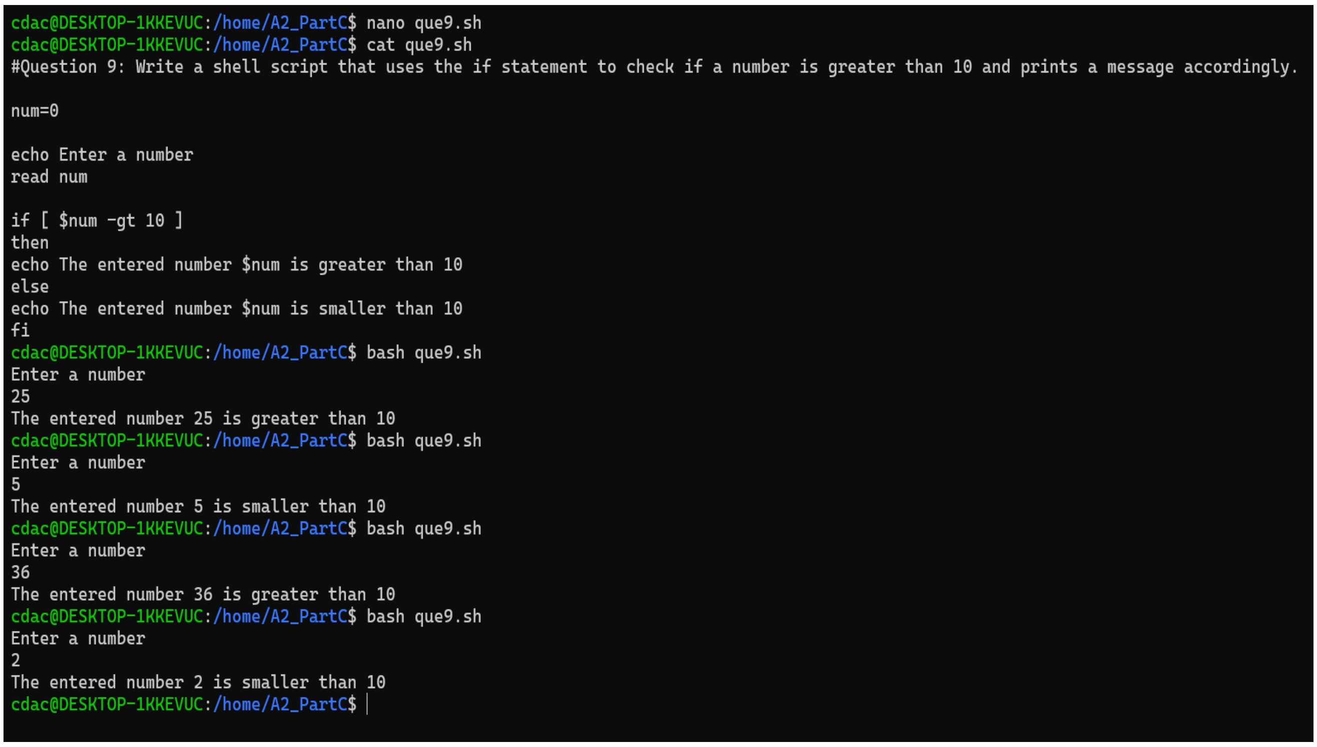
```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que7.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que7.sh
#Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.

num=1
while [ $num -lt 6 ]
do
echo $num
num=`expr $num + 1`
done
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que7.sh
1
2
3
4
5
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$
```

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".



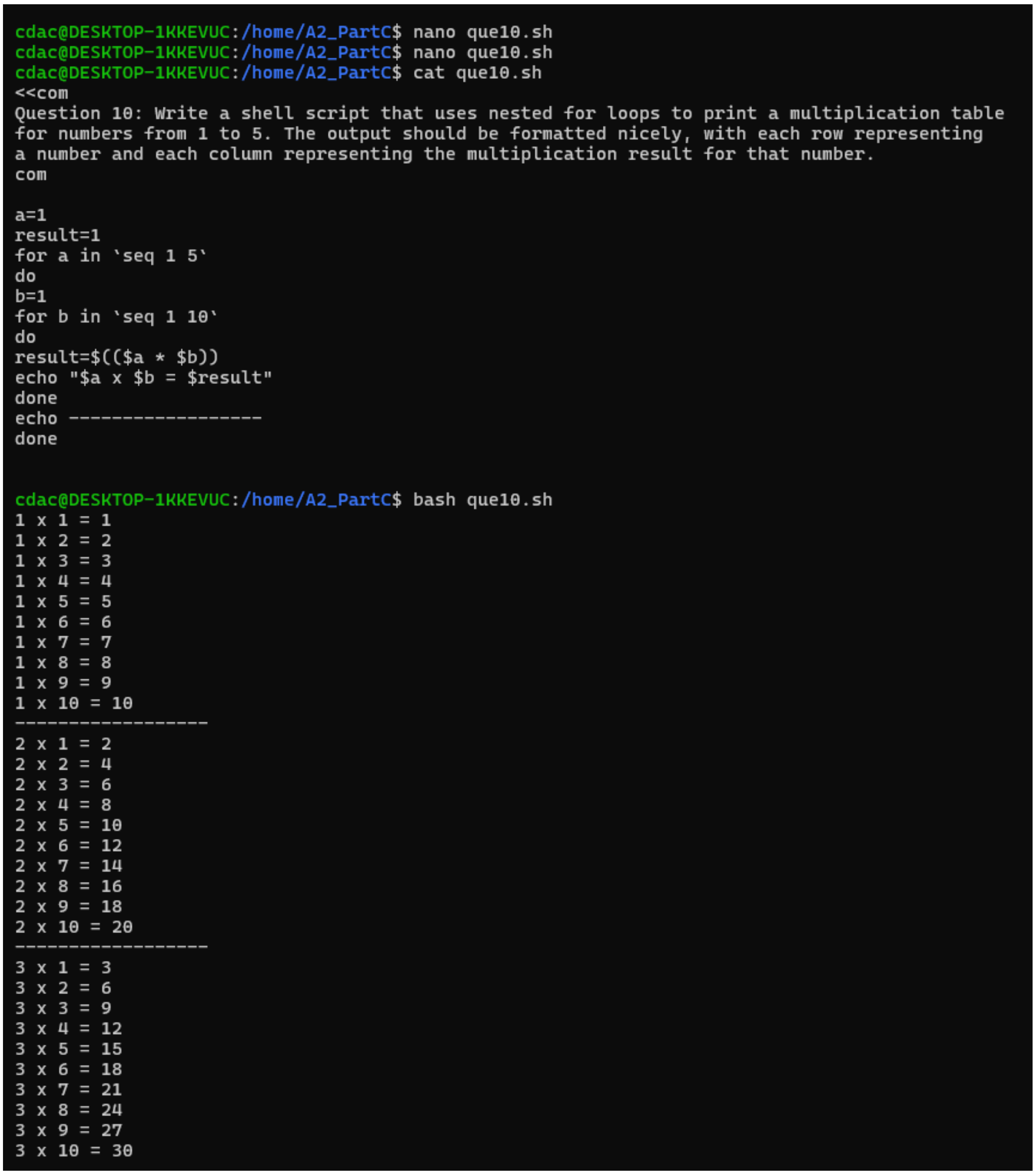
```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que8.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que8.sh
#Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory.
#If it does, print "File exists", otherwise, print "File does not exist".

if [ "find file.txt" == "file.txt" ]
then
echo File exits
else
echo File does not exist
fi
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que8.sh
File does not exist
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ touch file.txt
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que8.sh
File does not exist
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que8.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que8.sh
#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 "file.txt" ]
then
echo File exits
else
echo File does not exist
fi
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que8.sh
File exits
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ rm file.txt
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que8.sh
File does not exist
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$
```

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.

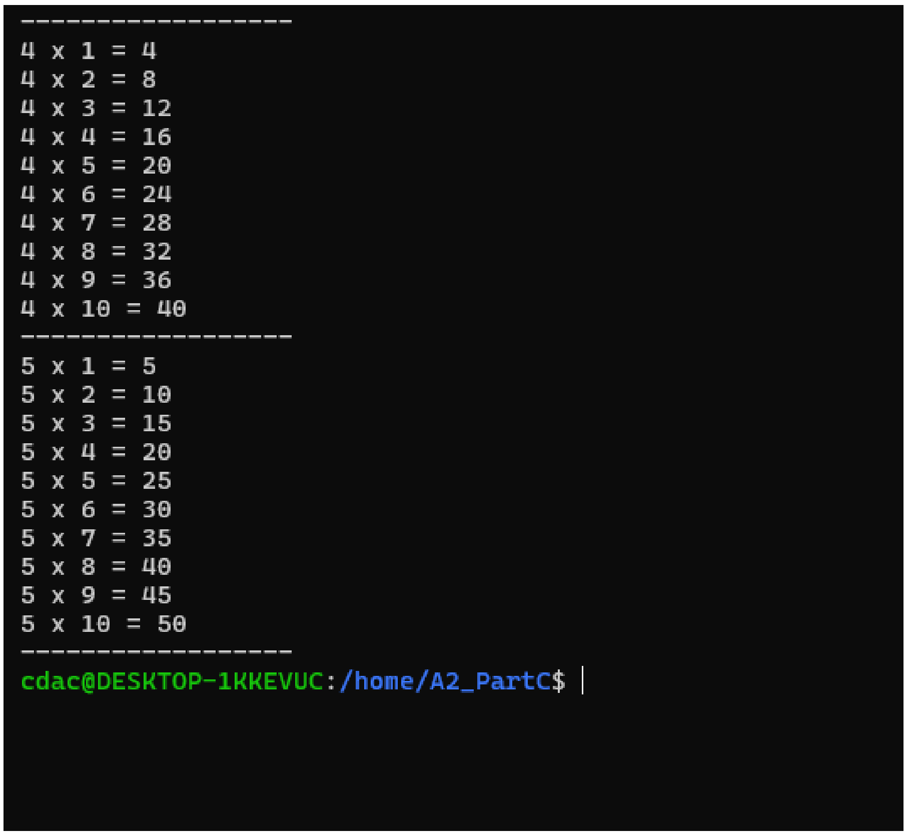


```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que9.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que9.sh
#Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.

num=0

echo Enter a number
read num

if [ $num -gt 10 ]
then
echo The entered number $num is greater than 10
else
echo The entered number $num is smaller than 10
fi
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que9.sh
Enter a number
25
The entered number 25 is greater than 10
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que9.sh
Enter a number
5
The entered number 5 is smaller than 10
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que9.sh
Enter a number
36
The entered number 36 is greater than 10
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que9.sh
Enter a number
2
The entered number 2 is smaller than 10
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$
```

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.

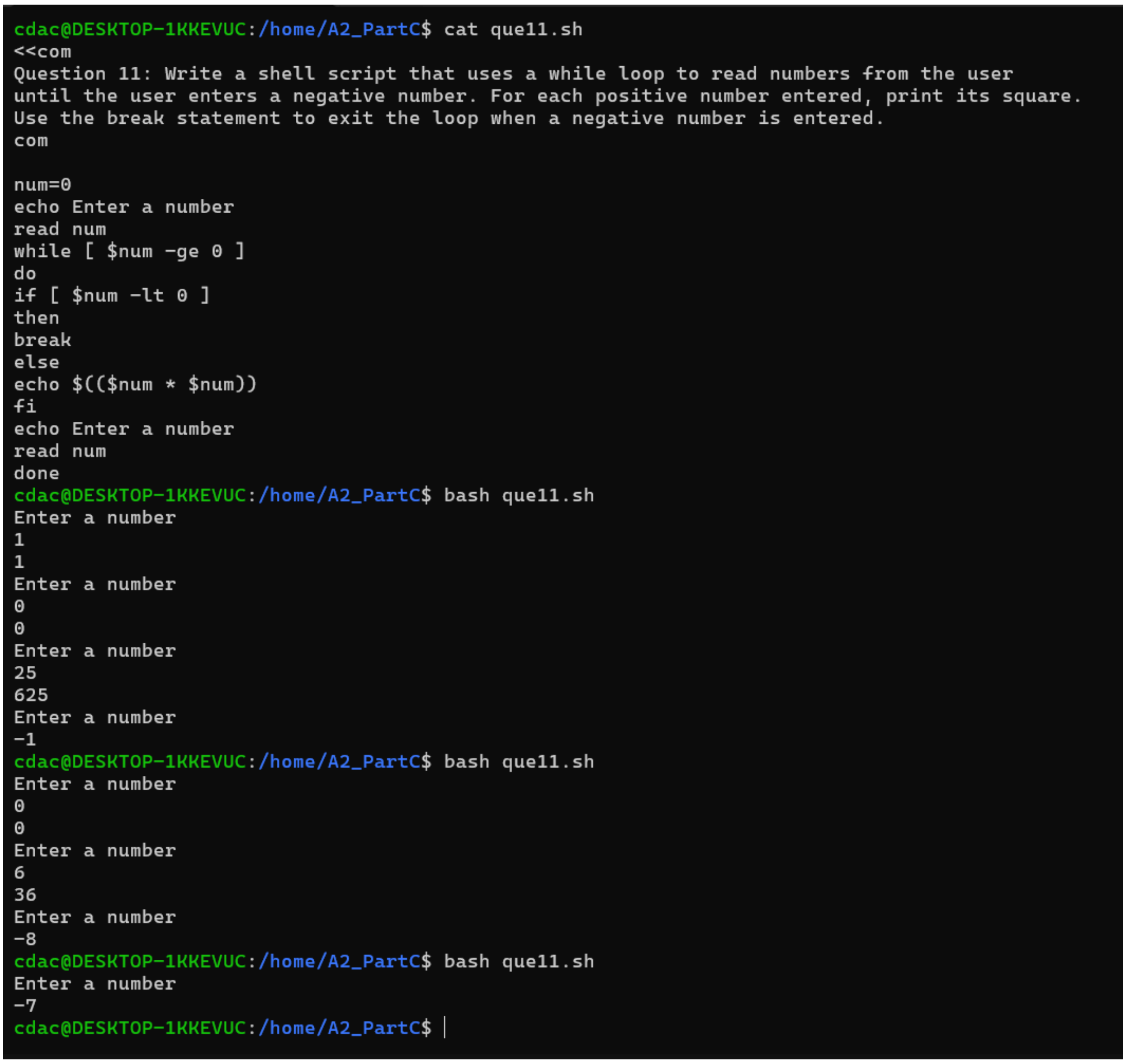


```
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que10.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ nano que10.sh
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ cat que10.sh
<<com
Question 10: Write a shell script that uses nested for loops to print a multiplication table
for numbers from 1 to 5. The output should be formatted nicely, with each row representing
a number and each column representing the multiplication result for that number.
com

a=1
result=1
for a in `seq 1 5`
do
b=1
for b in `seq 1 10`
do
result=$(($a * $b))
echo "$a x $b = $result"
done
echo ------------------
done


cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que10.sh
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
------------------
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
------------------
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```



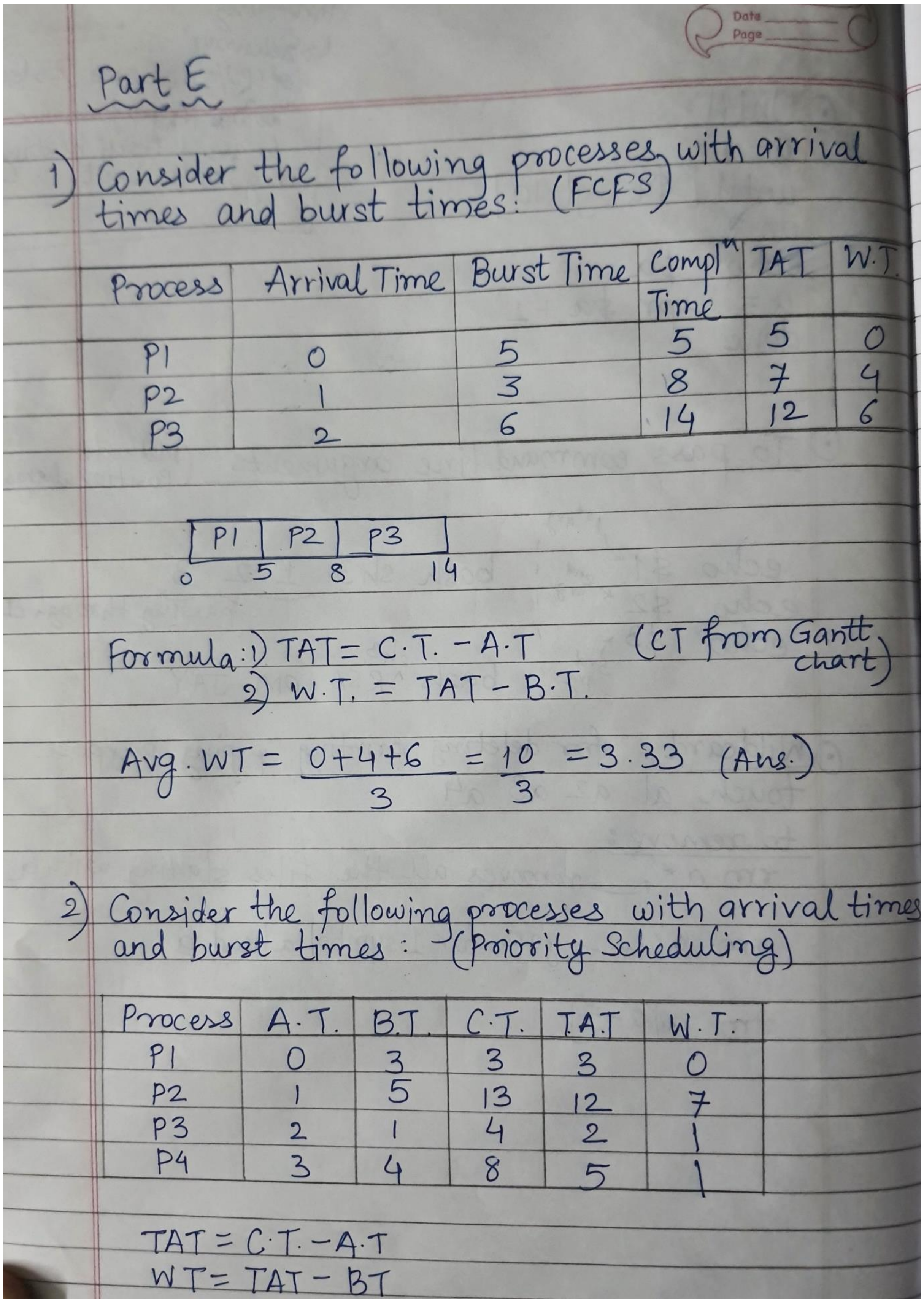
```
------------------
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40
------------------
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
------------------
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$
```

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-1KKEVUC:/home/A2_PartC$ cat que11.sh
<<com
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.
com

num=0
echo Enter a number
read num
while [ $num -ge 0 ]
do
if [ $num -lt 0 ]
then
break
else
echo $(($num * $num))
fi
echo Enter a number
read num
done
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que11.sh
Enter a number
1
1
Enter a number
0
0
Enter a number
25
625
Enter a number
-1
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que11.sh
Enter a number
0
0
Enter a number
6
36
Enter a number
-8
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ bash que11.sh
Enter a number
-7
cdac@DESKTOP-1KKEVUC:/home/A2_PartC$ |
```

Part E.



Date ____
Page ____

Part E

1) Consider the following processes with arrival times and burst times: (FCFS)

Process	Arrival Time	Burst Time	Compl^n Time	TAT	W.T.
P1	0	5	5	5	0
P2	1	3	8	7	4
P3	2	6	14	12	6

```
| P1 | P2 | P3 |
0    5    8    14
```

Formula: 1) TAT = C.T. − A.T (CT from Gantt chart)
2) W.T. = TAT − B.T.

$$\text{Avg. WT} = \frac{0+4+6}{3} = \frac{10}{3} = 3.33 \text{ (Ans.)}$$

2) Consider the following processes with arrival times and burst times: (Priority Scheduling)

Process	A.T.	B.T.	C.T.	TAT	W.T.
P1	0	3	3	3	0
P2	1	5	13	12	7
P3	2	1	4	2	1
P4	3	4	8	5	1

TAT = C.T. − A.T
WT = TAT − BT

Date
Page

Ans:

P1	P3	P4	P2

0 3 4 8 13

$$\text{Avg TAT} = \frac{3+12+2+5}{4} = \frac{22}{4} = 5.5$$

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

Process	A.T.	B.T.	Priority	C.T.	TAT	WT
P1	0	6	3	12	12	6
P2	1	4	1	5	4	0
P3	2	7	4	19	17	10
P4	3	2	2	7	4	2

Calc. avg w.t. using Priority Scheduling.

Ans:

P1	P2	P4	P1	P3

0 1 5 7 12 19

$$\text{Avg. WT} = \frac{6+0+10+2}{4} = \frac{18}{4} = 4.5$$

Rough Work

1	~~6~~5
2	4
3	7
4	2

4) Consider the following processes with arrival times and burst times, and the time quantum for Round Robin Scheduling is 2 units.

Process	A.T.	B.T.	C.T.	TAT
P1	0	4	10	10
P2	1	5	14	13
P3	2	2	6	4
P4	3	3	13	10

Calc. the avg. TAT using Round Robin scheduling

→

P1	P2	P3	P4	P1	P2	P4	P2

```
0    2    4    6    8    10   12   13   14
```

$$\text{Avg TAT} = \frac{10+13+4+10}{4} = \frac{37}{4} = 9.25 \text{ (Ans.)}$$

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?

→ 1) Before fork():
Parent has $x=5$

2) After fork():
- The child gets a separate copy of $x=5$.
- Both parent and child continue execution from the next instruction after fork().

3) Incrementing x in both processes:
- Parent increments its own x as $x=6$
- Child increments its own x as $x=6$

Since both processes have independent memory, changes in one do not affect the other.

Final values of x:
In parent process: $x=6$
In child process: $x=6$