Tejas Rajendra Patil

Center-JUHU

Assignment 2

Part A

What will the following commands do?

- echo "Hello, World!"

  ➔ Prints Hello, World! to the terminal.

- name="Productive"

  ➔ Creates a variable name and assigns it the value Productive .

- touch file.txt

  ➔ Creates an empty file named file.txt or updates its timestamp if it already exists

- ls -a

  ➔ Lists all files and directories in the current directory, including hidden ones (those starting with .)

- rm file.txt

  ➔ Removes the file file.txt permanently.

- cp file1.txt file2.txt

  ➔ Copies file1.txt to file2.txt . If file2.txt exists, it will be overwritten.

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

  ➔ Moves file.txt to the specified directory.

- chmod 755 script.sh

  ➔ Grants the owner full permissions (read, write, execute) and gives others read and execute permissions on script.sh

- grep "pattern" file.txt

  ➔ Searches for occurrences of "pattern" in file.txt and prints matching lines.

- kill PID

  ➔ Terminates the process with the specified Process ID (PID).

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

  ➔

  1. Creates a directory mydir
  2. Changes into mydir
  3. Creates an empty file file.txt

4. Writes "Hello, World!" into file.txt
5. Displays the contents of file.txt

- ls -l | grep ".txt"

  ➔ Lists files in long format and filters only those containing ". Txt" in their names.

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

  ➔ Concatenates file1.txt and file2.txt , sorts them, and removes duplicate lines.

- ls -l | grep "^d"

  ➔ Lists directories (entries starting with d in long format output).

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

  ➔ Searches for "pattern" recursively in all files under /path/to/directory/

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

  ➔ Concatenates file1.txt and file2.txt , sorts them, and displays only duplicate lines.

- chmod 644 file.txt

  ➔ Grants the owner read and write permissions, while others get read-only access to file.txt .

- cp -r source_directory destination_directory

  ➔ Recursively copies source_directory to destination_directory , preserving contents.

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

  ➔ Finds all .txt files in /path/to/search and its subdirectories.

- chmod u+x file.txt

  ➔ Gives the owner (u) execute permission on file.txt .

- echo $PATH

  ➔ Displays the system's PATH environment variable, listing directories where executable files are searched for.

Part B - Identify True or False

1. **True** - ls is used to list files and directories in a directory.

2. **True** - mv is used to move files and directories.

3. **False** - cd is used to change directories, not copy files and directories.

4. **True** - pwd stands for "print working directory" and displays the current directory.

5. **True** - grep is used to search for patterns in files.

6. **True** - chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.

7. **True** - mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1 if directory1 does not exist.

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

1. **Incorrect** - chmodx is not a valid command. The correct command to change file permissions is chmod .

2. **Incorrect** - cpy is not a valid command. The correct command to copy files and directories is cp .

3. **Incorrect** - mkfile is not a standard Linux command. To create a new file, use filename .

4. **Incorrect** - touch catx is not a valid command. The correct command to concatenate files is cat .

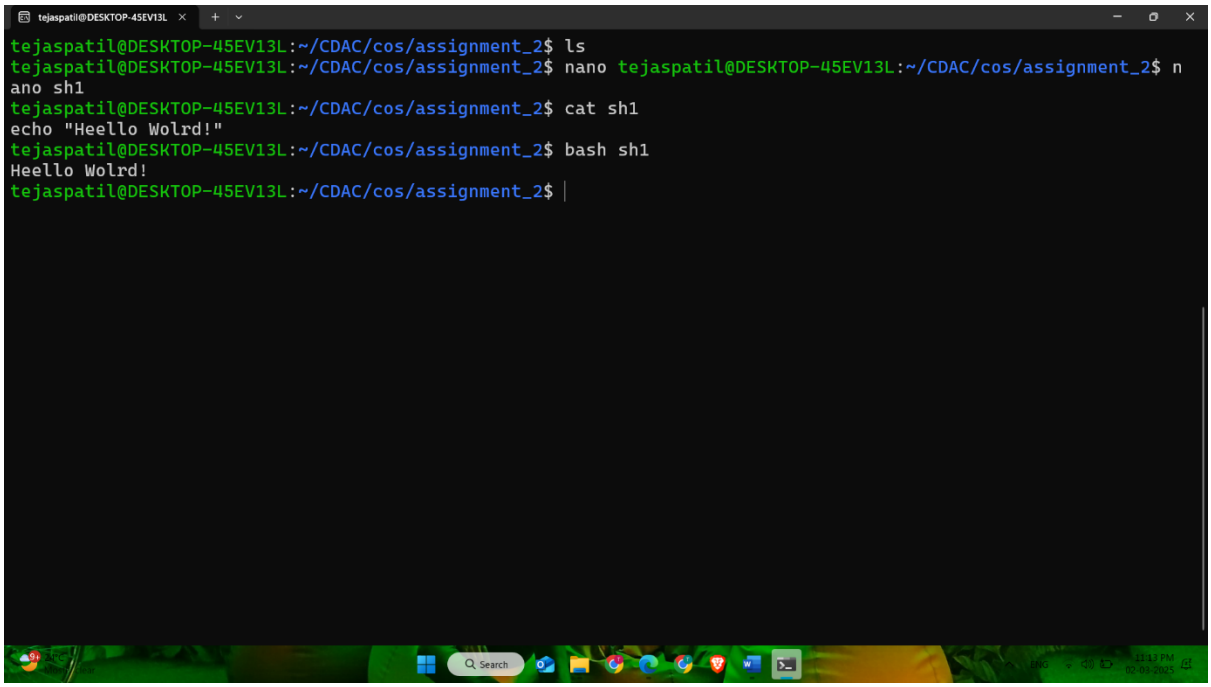
5. **Incorrect** - rn is not a valid command. To rename files, use the mv command (oldname newname).

Part C - Shell Scripting Questions

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

#!/bin/bash

echo "Hello, World!"

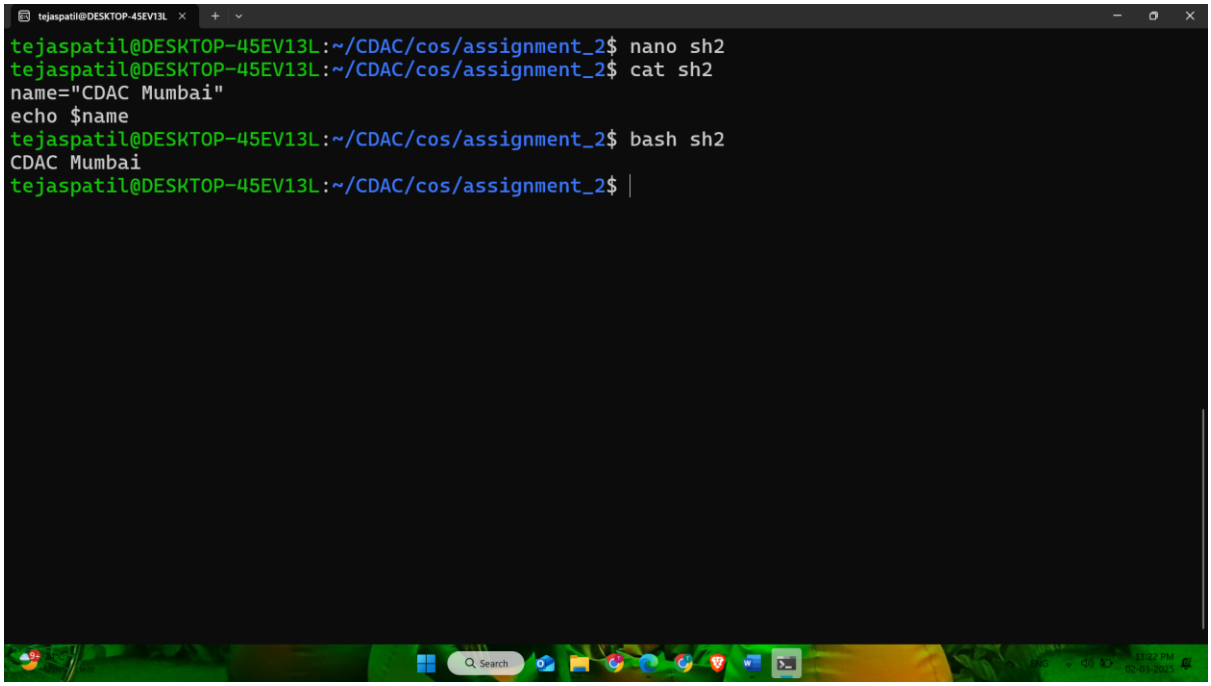


Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.

#!/bin/bash

name="CDAC Mumbai"

echo "$name"

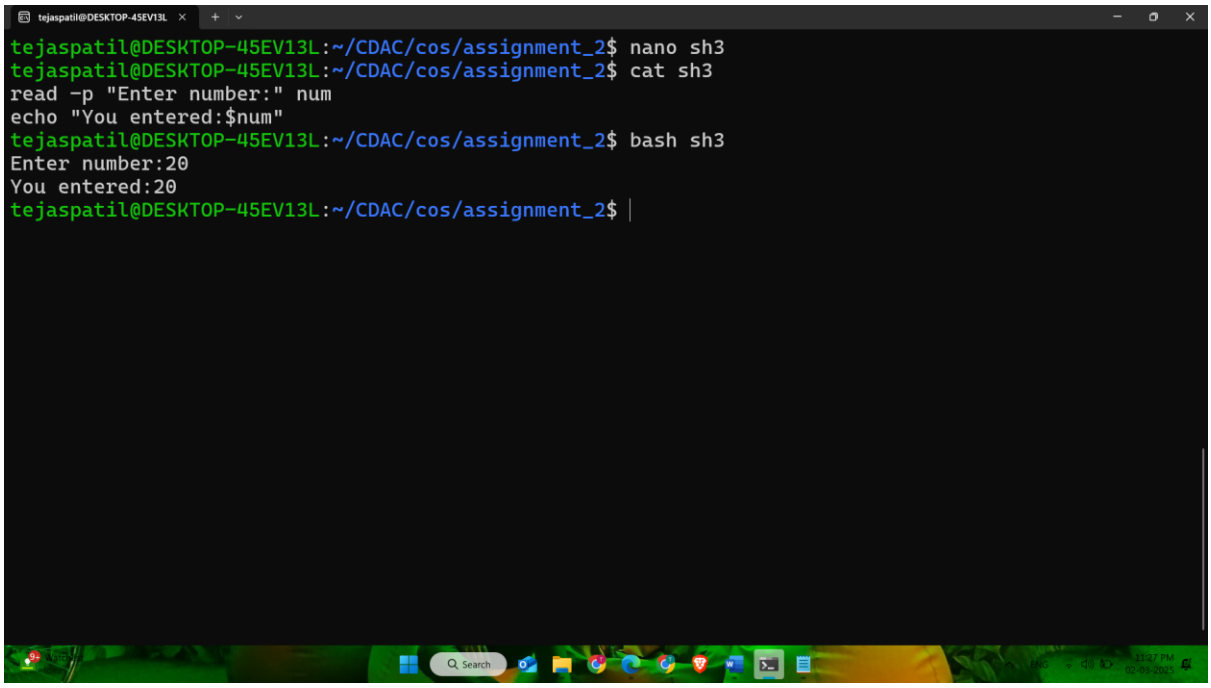


Question 3: Write a shell script that takes a number as input from the user and prints it.

#!/bin/bash

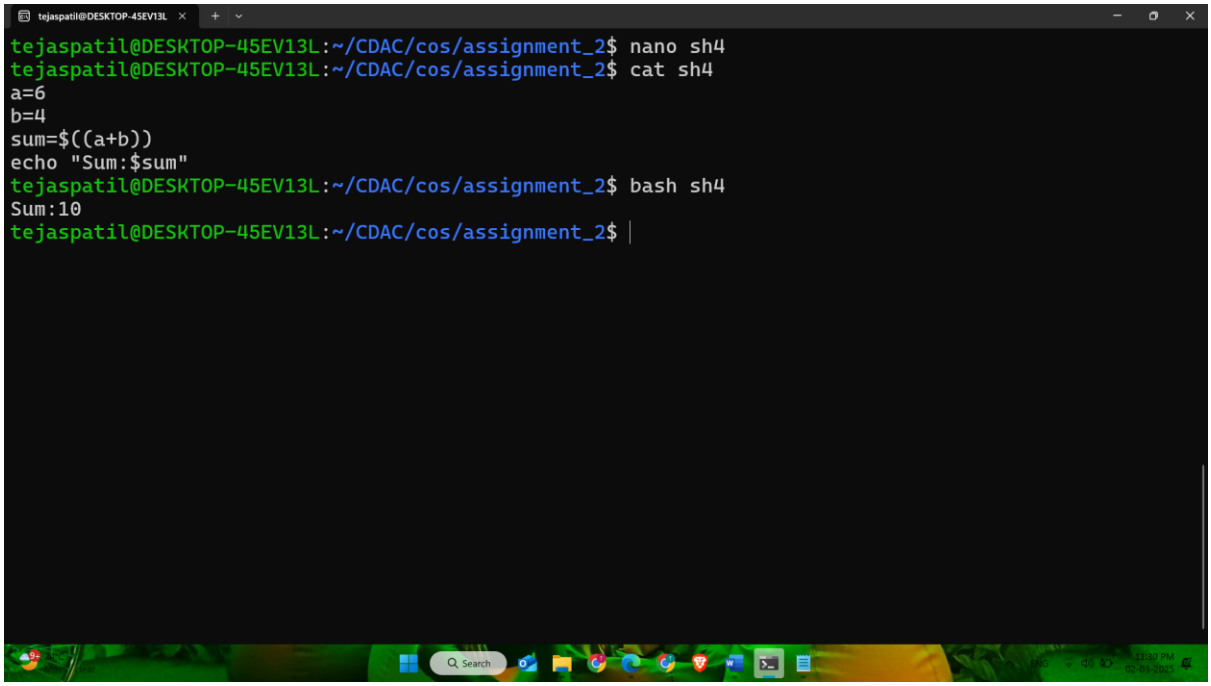
read-p "Enter a number: " num

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.

```
#!/bin/bash
a=5
b=3
sum=$((a + b))
echo "Sum: $sum"
```



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

```
#!/bin/bash
read-p "Enter a number: " num
if ((num % 2 == 0)); then
echo "Even"
else
echo "Odd"
fi
```

```
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ nano sh5
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ cat sh5
 read -p "Enter a number: " num
 if ((num % 2 == 0)); then
 echo "Even"
 else
 echo "Odd"
 fi
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ bash sh5
Enter a number: 14
Even
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ bash sh5
Enter a number: 15
Odd
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$
```

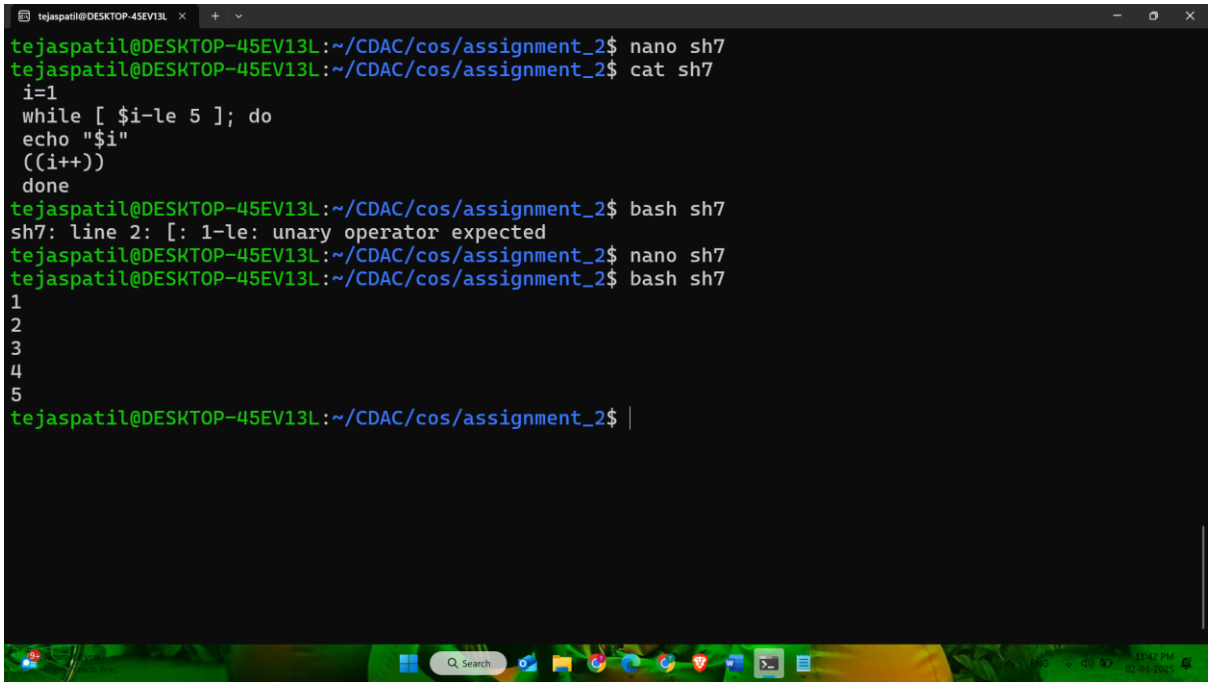
Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

```
#!/bin/bash
 for i in {1..5}; do
 echo "$i"
 done
```

```
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ nano sh6
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ cat sh6
 for i in {1..5}; do
 echo "$i"
 done
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ bash sh6
1
2
3
4
5
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$
```

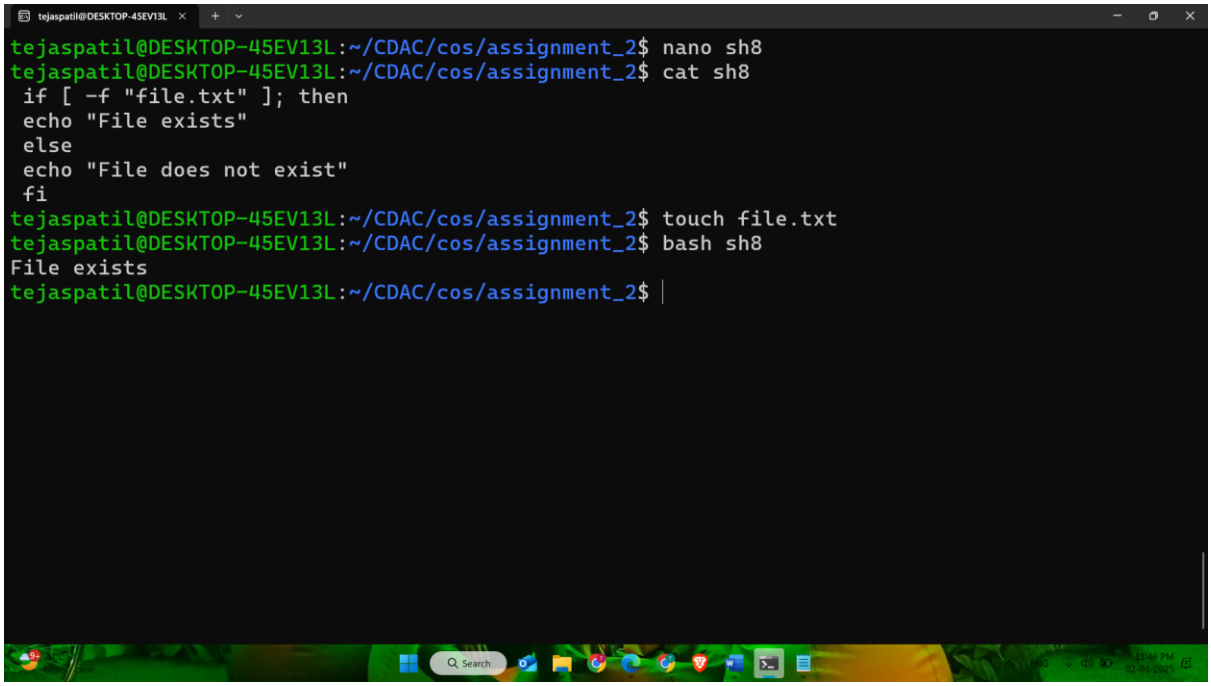
Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.

```
#!/bin/bash
 i=1
 while [ $i-le 5 ]; do
 echo "$i"
 ((i++))
 Done
```



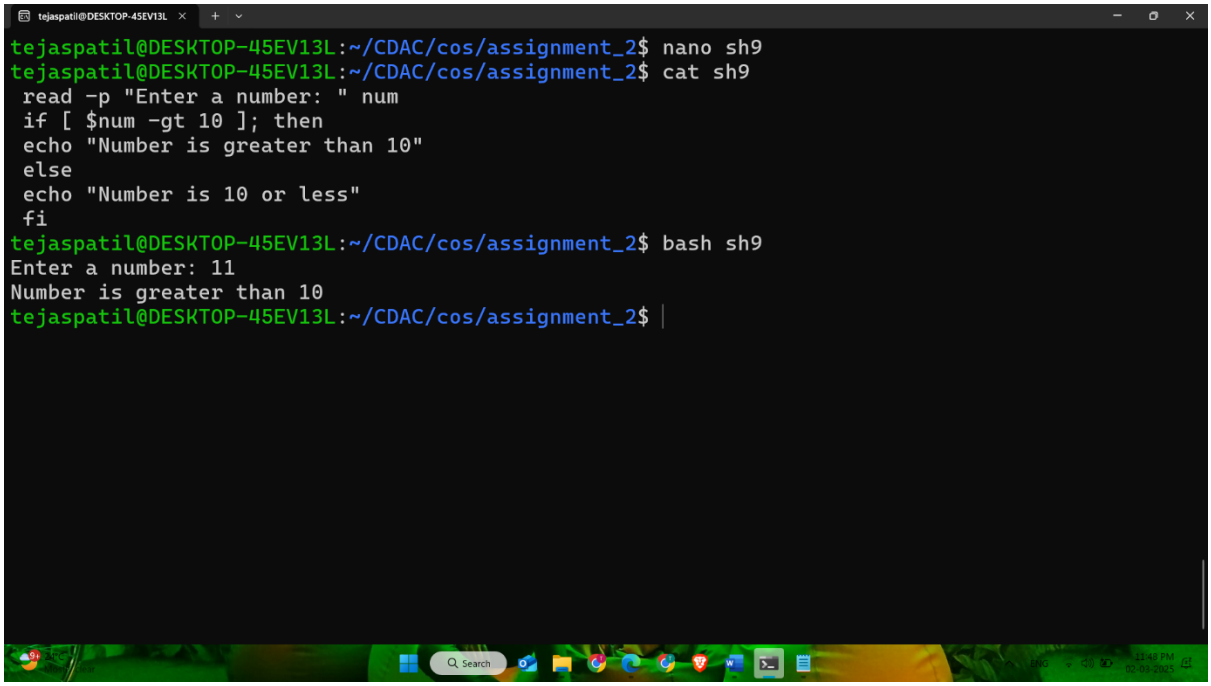
Question 8: Write a shell script that checks if a file named "file. Txt" exists in the current directory.

```
#!/bin/bash
 if [-f "file.txt" ]; then
 echo "File exists"
 else
 echo "File does not exist"
 fi
```



Question 9: Write a shell script that checks if a number is greater than 10 and prints a message accordingly.

```
#!/bin/bash
read-p "Enter a number: " num
if [ $num-gt 10 ]; then
echo "Number is greater than 10"
else
echo "Number is 10 or less"
fi
```

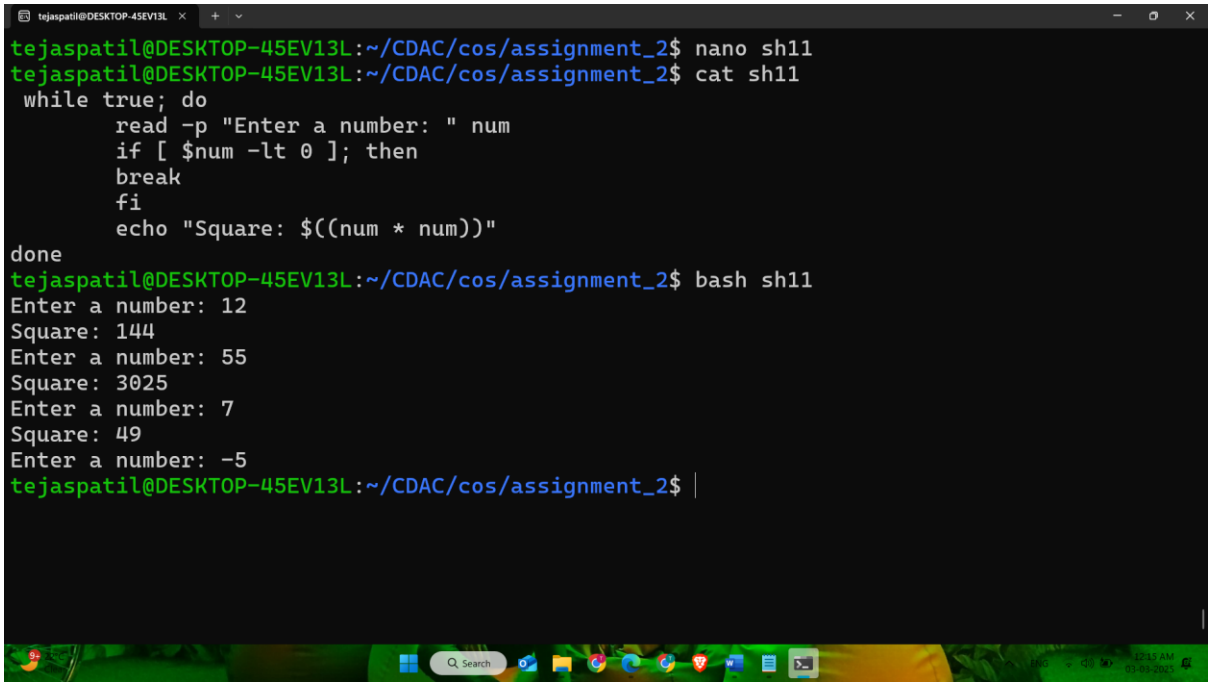


Question 10: Write a shell script that prints a multiplication table for numbers from 1 to 5.

```
#!/bin/bash
 for i in {1..5}; do
 for j in {1..5}; do
 printf "%4d" $((i * j))
 done
 echo
 done
```

Question 11: Write a shell script that reads numbers from the user until a negative number is entered.

```
while true; do
    read -p "Enter a number: " num
    if [ $num -lt 0 ]; then
    break
    fi
    echo "Square: $((num * num))"
done
```


tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ nano sh11
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ cat sh11
while true; do
read -p "Enter a number: " num
if [$num -lt 0]; then
break
fi
echo "Square: $((num * num))"
done
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$ bash sh11
Enter a number: 12
Square: 144
Enter a number: 55
Square: 3025
Enter a number: 7
Square: 49
Enter a number: -5
tejaspatil@DESKTOP-45EV13L:~/CDAC/cos/assignment_2$