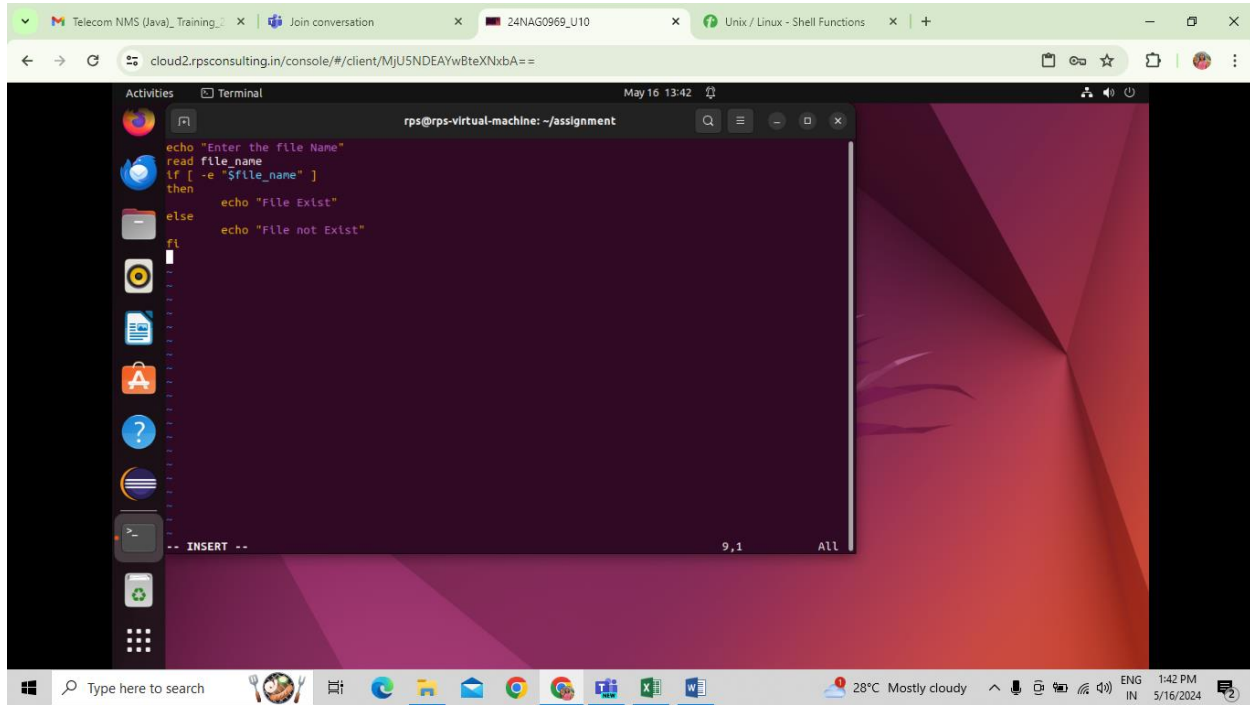


## ASSIGNMENT -05

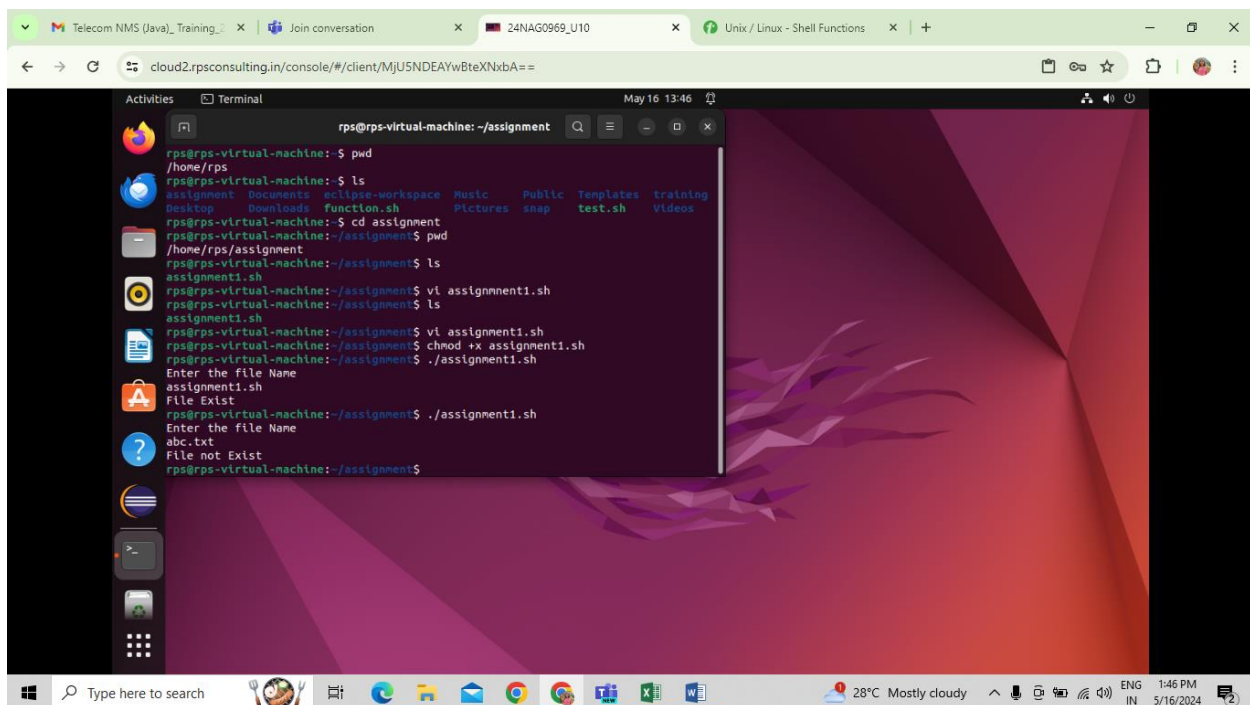
1] Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".



The screenshot shows a terminal window titled "rps@rps-virtual-machine: ~/assignment". The script content is as follows:

```
echo "Enter the file Name"
read file_name
if [ -e "$file_name" ]
then
    echo "File Exist"
else
    echo "File not Exist"
fi
```

The terminal is running on a Linux desktop environment with a purple and red geometric wallpaper. The system tray at the bottom shows the date as May 16, 13:42, and the temperature as 28°C.

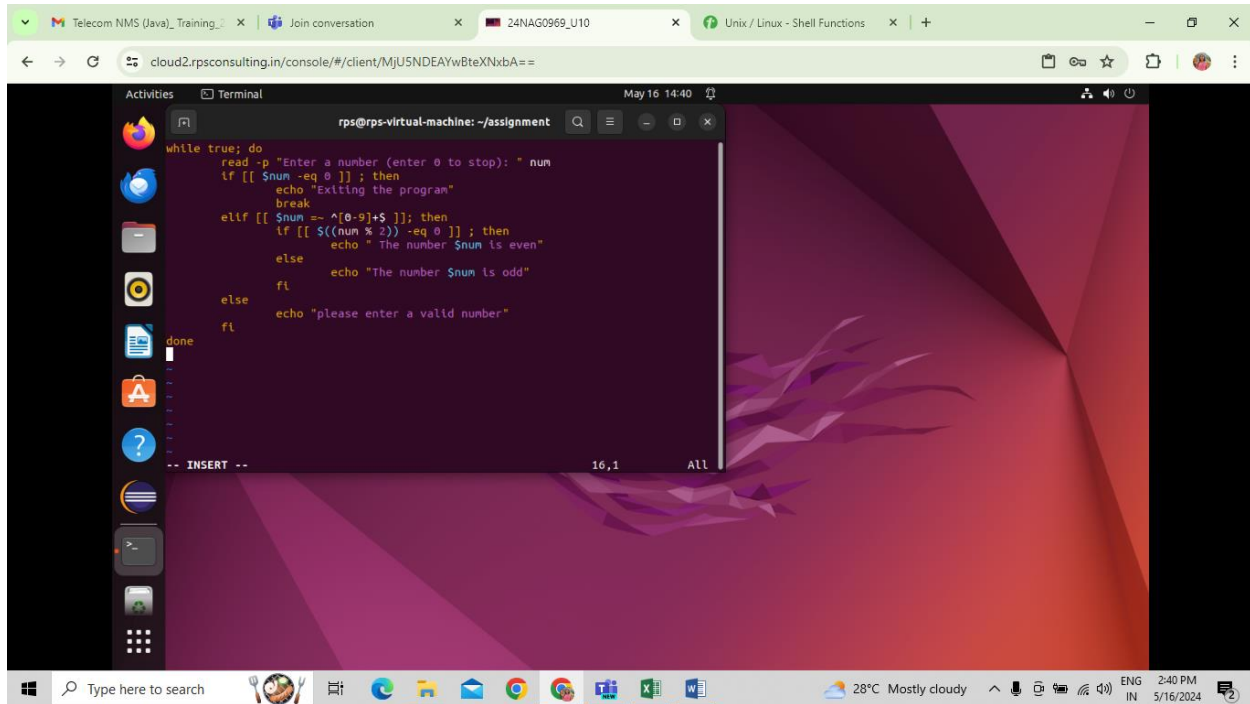


The screenshot shows the same terminal window after several commands have been executed. The output is as follows:

```
rps@rps-virtual-machine:~$ pwd
/home/rps
rps@rps-virtual-machine:~$ ls
assignment  Desktop  Downloads  function.sh  Music  Public  Templates  training
rps@rps-virtual-machine:~$ cd assignment
rps@rps-virtual-machine:~/assignment$ pwd
/home/rps/assignment
rps@rps-virtual-machine:~/assignment$ ls
assignment1.sh
rps@rps-virtual-machine:~/assignment$ vi assignment1.sh
rps@rps-virtual-machine:~/assignment$ ls
assignment1.sh
rps@rps-virtual-machine:~/assignment$ vi assignment1.sh
rps@rps-virtual-machine:~/assignment$ chmod +x assignment1.sh
rps@rps-virtual-machine:~/assignment$ ./assignment1.sh
Enter the file Name
assignment1.sh
File Exist
rps@rps-virtual-machine:~/assignment$ ./assignment1.sh
Enter the file Name
abc.txt
File not Exist
rps@rps-virtual-machine:~/assignment$
```

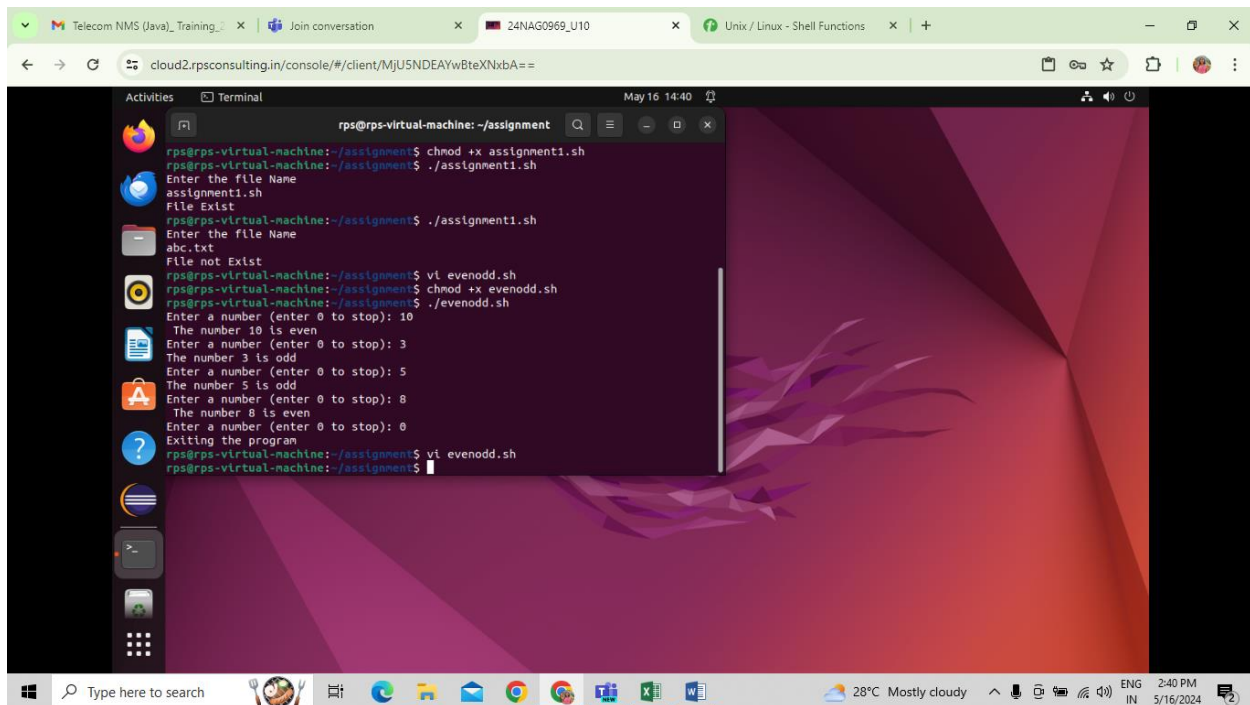
The terminal shows the user navigating to the assignment directory, listing files, editing assignment1.sh, and running it twice. The first run with "assignment1.sh" as input prints "File Exist", and the second run with "abc.txt" as input prints "File not Exist". The system tray at the bottom shows the date as May 16, 1:46 PM.

2] Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even.



The screenshot shows a terminal window titled 'rps@rps-virtual-machine: ~/assignment'. The user has created a file named 'assignment' and is viewing its contents. The script is a shell script that uses a 'while true' loop to repeatedly prompt the user for a number. It checks if the number is 0 to break the loop, or if it's a valid integer. If valid, it checks if the number is even or odd and prints the result. If not a valid integer, it prompts the user to enter a valid number.

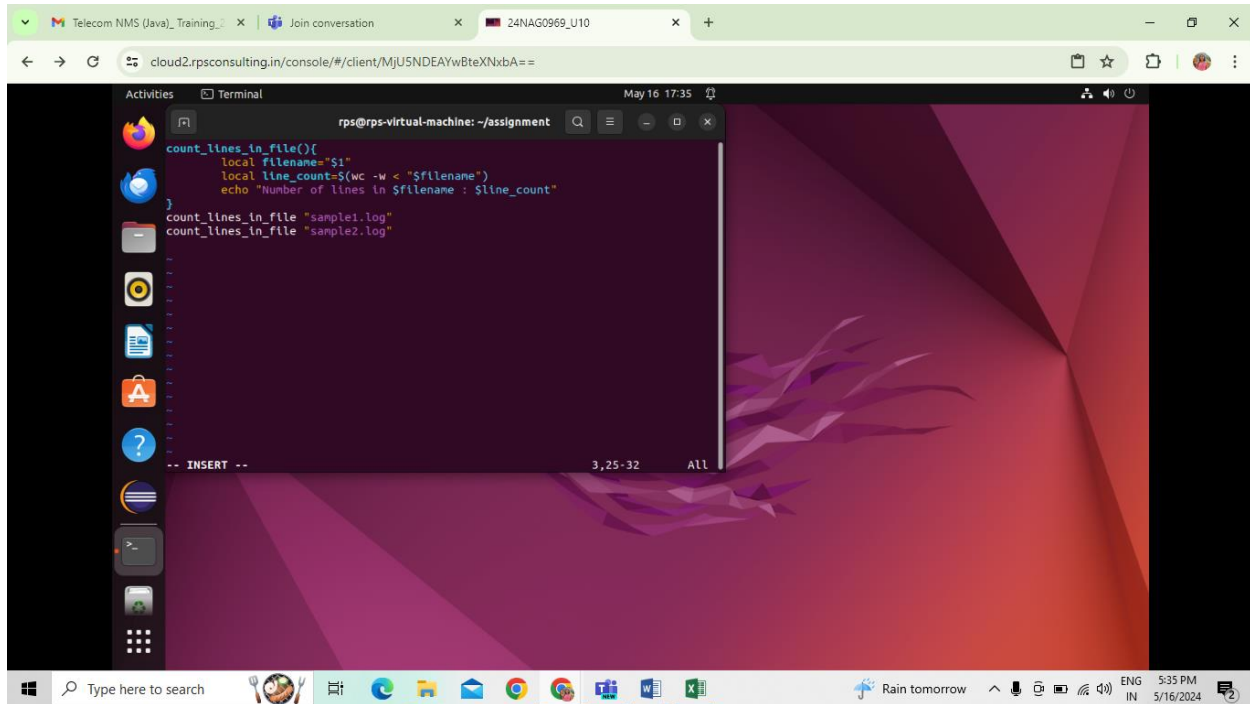
```
while true; do
    read -p "Enter a number (enter 0 to stop): " num
    if [[ $num -eq 0 ]]; then
        echo "Exiting the program"
        break
    elif [[ $num =~ ^[0-9]+$ ]]; then
        if [[ $((num % 2)) -eq 0 ]]; then
            echo "The number $num is even"
        else
            echo "The number $num is odd"
        fi
    else
        echo "please enter a valid number"
    fi
done
```



The screenshot shows the same terminal window after the script has been executed. The user has made the script executable with 'chmod +x assignment1.sh' and then run it with './assignment1.sh'. The script prompts for a file name, and the user enters 'assignment1.sh'. The script then prompts for a number, and the user enters '10', '3', '5', and '8'. The script prints the results: 'The number 10 is even', 'The number 3 is odd', 'The number 5 is odd', and 'The number 8 is even'. Finally, the user enters '0' to exit the program.

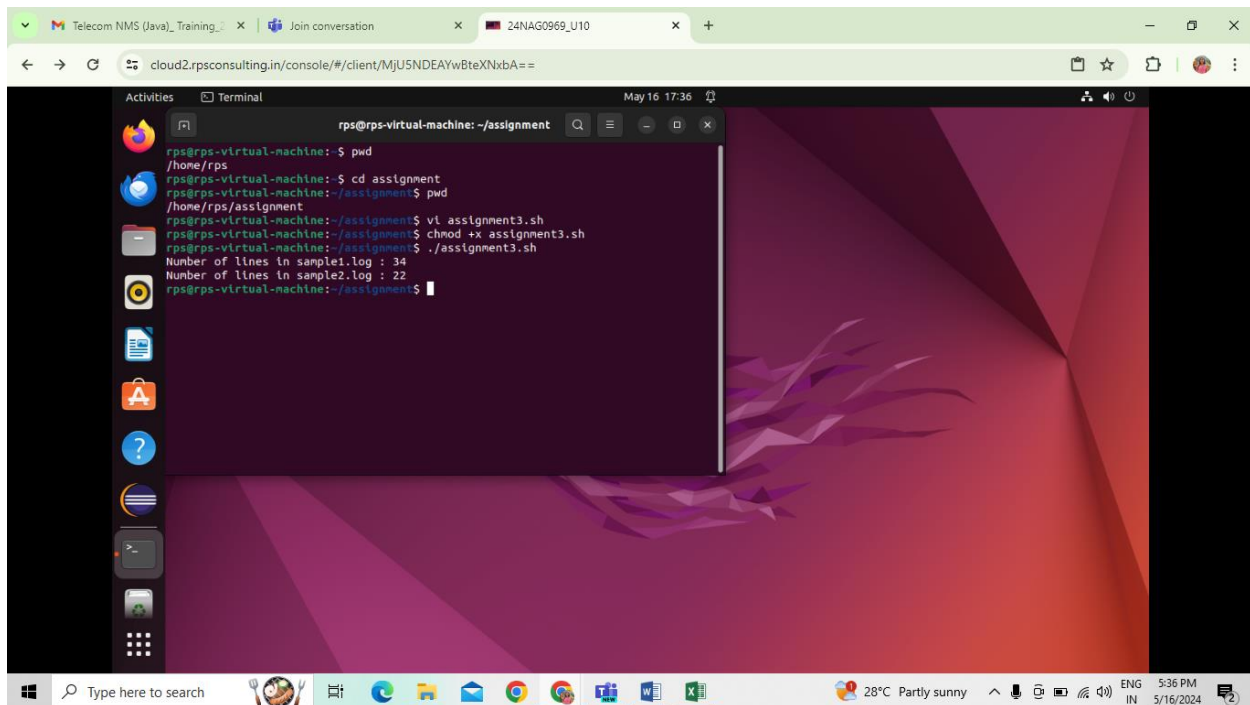
```
rps@rps-virtual-machine:~/assignment$ chmod +x assignment1.sh
rps@rps-virtual-machine:~/assignment$ ./assignment1.sh
Enter the file Name
assignment1.sh
File Exist
rps@rps-virtual-machine:~/assignment$ ./assignment1.sh
Enter the file Name
abc.txt
File not Exist
rps@rps-virtual-machine:~/assignment$ vi evenodd.sh
rps@rps-virtual-machine:~/assignment$ chmod +x evenodd.sh
rps@rps-virtual-machine:~/assignment$ ./evenodd.sh
Enter a number (enter 0 to stop): 10
The number 10 is even
Enter a number (enter 0 to stop): 3
The number 3 is odd
Enter a number (enter 0 to stop): 5
The number 5 is odd
Enter a number (enter 0 to stop): 8
The number 8 is even
Enter a number (enter 0 to stop): 0
Exiting the program
rps@rps-virtual-machine:~/assignment$ vi evenodd.sh
rps@rps-virtual-machine:~/assignment$
```

3] Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.



The screenshot shows a terminal window titled "rps@rps-virtual-machine: ~/assignment". The terminal displays a function definition for `count_lines_in_file` and its execution. The function takes a filename as an argument and prints the number of lines in the file. The function is called with "sample1.log" and "sample2.log".

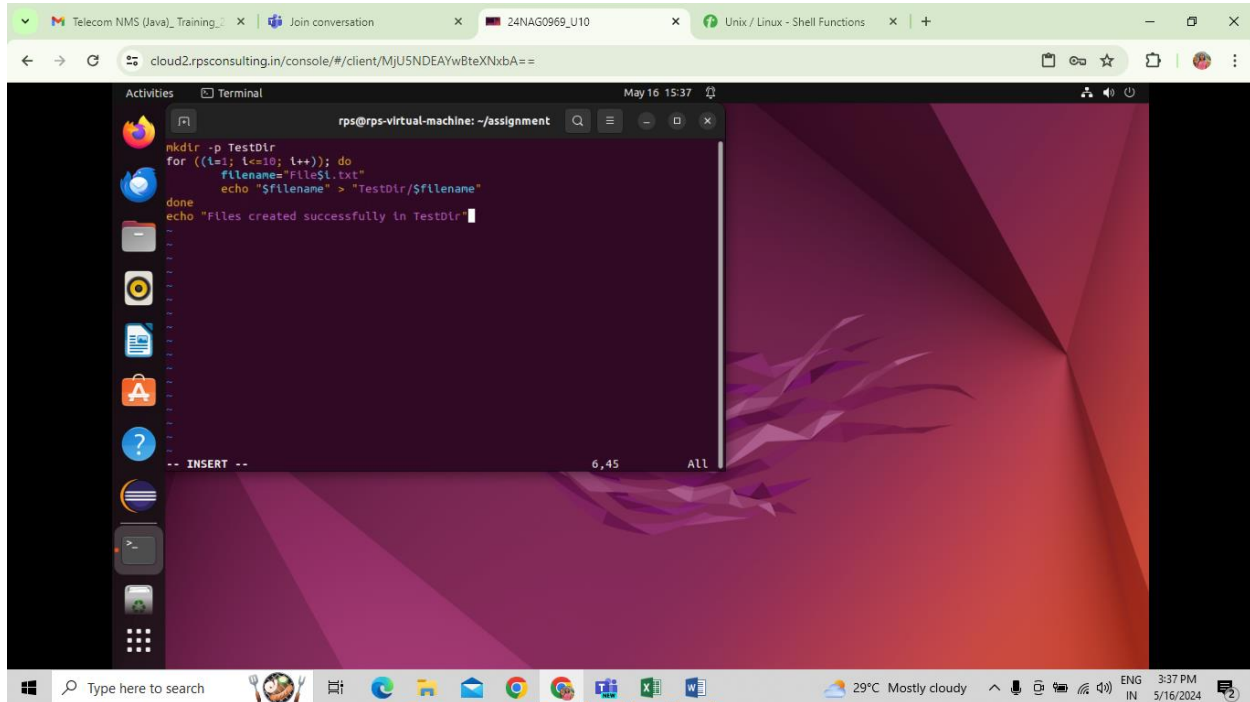
```
rps@rps-virtual-machine: ~/assignment
count_lines_in_file(){
    local filename="$1"
    local line_count=$(wc -w < "$filename")
    echo "Number of lines in $filename : $line_count"
}
count_lines_in_file "sample1.log"
count_lines_in_file "sample2.log"
```



The screenshot shows a terminal window titled "rps@rps-virtual-machine: ~/assignment". The terminal displays the execution of a script named `assignment3.sh`. The script is run with `vt assignment3.sh` and `chmod +x assignment3.sh`. The output shows the number of lines in `sample1.log` and `sample2.log`.

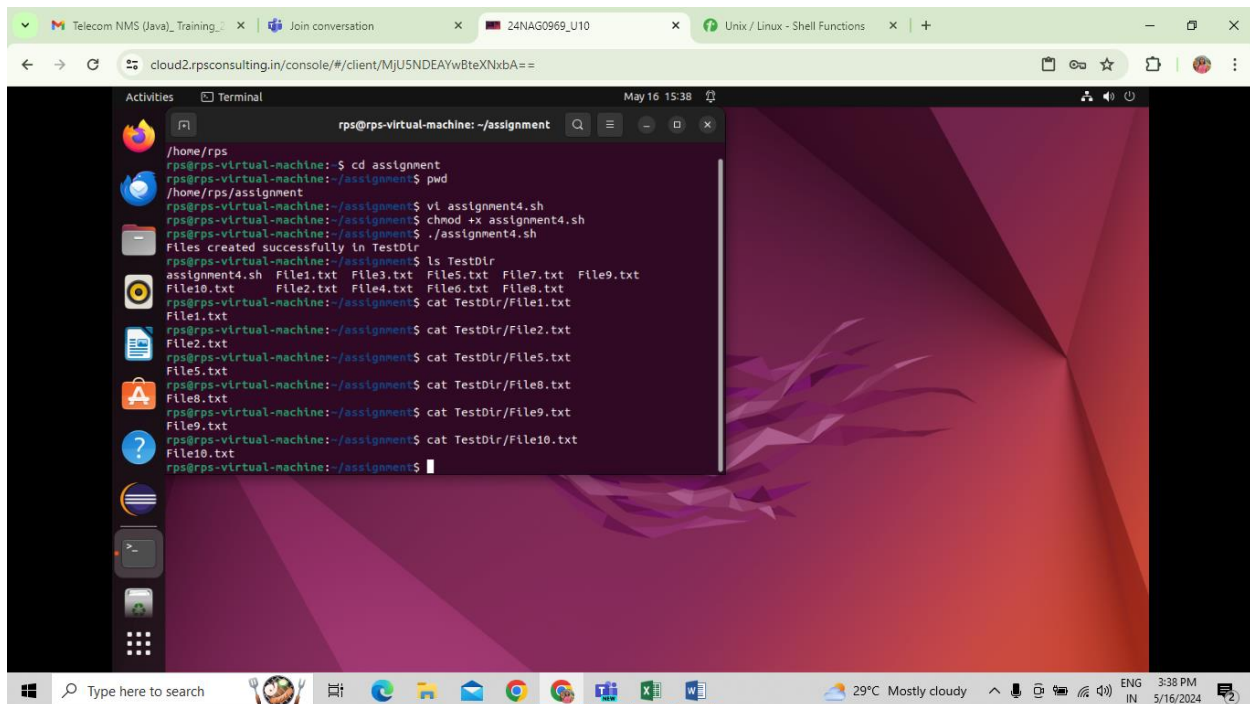
```
rps@rps-virtual-machine: $ pwd
/home/rps
rps@rps-virtual-machine: $ cd assignment
rps@rps-virtual-machine:~/assignment$ pwd
/home/rps/assignment
rps@rps-virtual-machine:~/assignment$ vt assignment3.sh
rps@rps-virtual-machine:~/assignment$ chmod +x assignment3.sh
rps@rps-virtual-machine:~/assignment$ ./assignment3.sh
Number of lines in sample1.log : 34
Number of lines in sample2.log : 22
rps@rps-virtual-machine:~/assignment$
```

4] Write a script that creates a directory named TestDir and inside it, creates ten files named File1.txt, File2.txt, ... File10.txt. Each file should contain its filename as its content (e.g., File1.txt contains "File1.txt").



The screenshot shows a terminal window titled "rps@rps-virtual-machine: ~/assignment". The user has executed a script that creates a directory named "TestDir" and ten files named "File1.txt" through "File10.txt". The output of the script is displayed in the terminal:

```
rps@rps-virtual-machine: ~/assignment
mkdir -p TestDir
for ((i=1; i<=10; i++)); do
    filename="File${i}.txt"
    echo "$filename" > "TestDir/$filename"
done
echo "Files created successfully in TestDir"
```



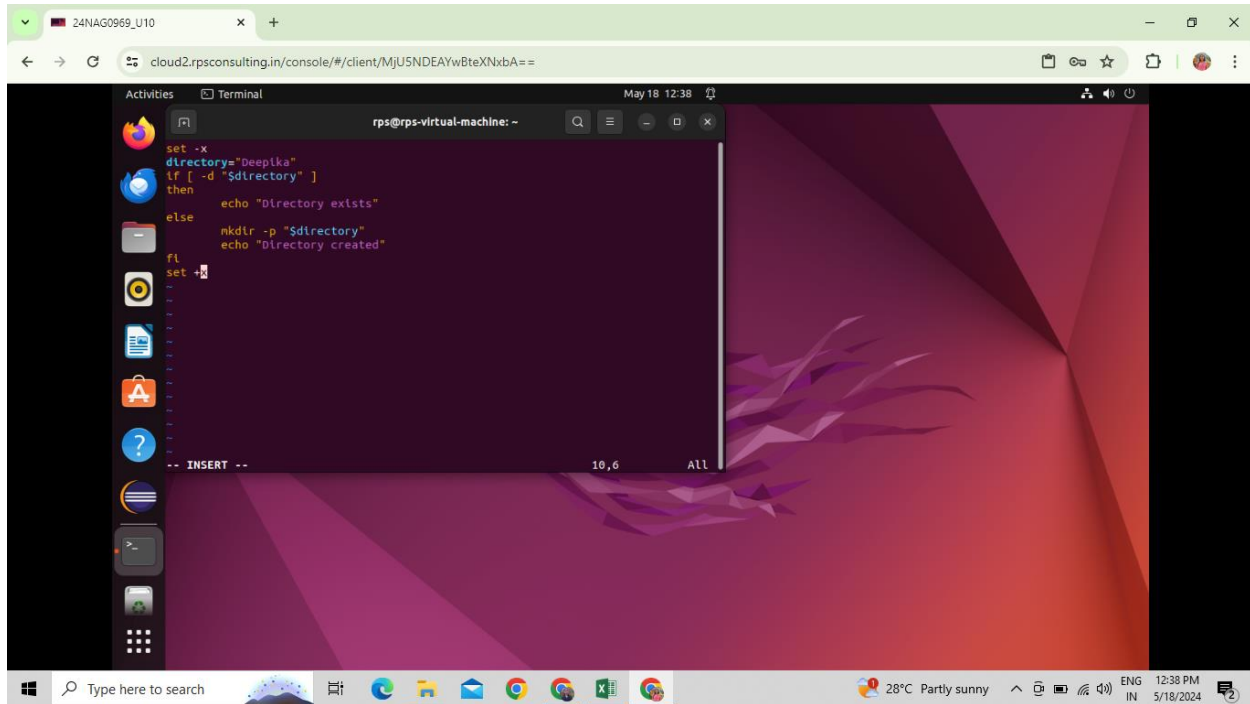
The screenshot shows the same terminal window with the user verifying the creation of the files. The user has navigated to the "assignment" directory and run the script "assignment4.sh". The output of the script is displayed in the terminal:

```
rps@rps-virtual-machine: ~/assignment
$ cd assignment
rps@rps-virtual-machine: ~/assignment$ pwd
/home/rps/assignment
rps@rps-virtual-machine: ~/assignment$ vl assignment4.sh
rps@rps-virtual-machine: ~/assignment$ chmod +x assignment4.sh
rps@rps-virtual-machine: ~/assignment$ ./assignment4.sh
Files created successfully in TestDir
rps@rps-virtual-machine: ~/assignment$ ls TestDir
assignment4.sh  File1.txt  File3.txt  File5.txt  File7.txt  File9.txt
File10.txt     File2.txt  File4.txt  File6.txt  File8.txt
rps@rps-virtual-machine: ~/assignment$ cat TestDir/File1.txt
File1.txt
rps@rps-virtual-machine: ~/assignment$ cat TestDir/File2.txt
File2.txt
rps@rps-virtual-machine: ~/assignment$ cat TestDir/File5.txt
File5.txt
rps@rps-virtual-machine: ~/assignment$ cat TestDir/File8.txt
File8.txt
rps@rps-virtual-machine: ~/assignment$ cat TestDir/File9.txt
File9.txt
rps@rps-virtual-machine: ~/assignment$ cat TestDir/File10.txt
File10.txt
rps@rps-virtual-machine: ~/assignment$
```

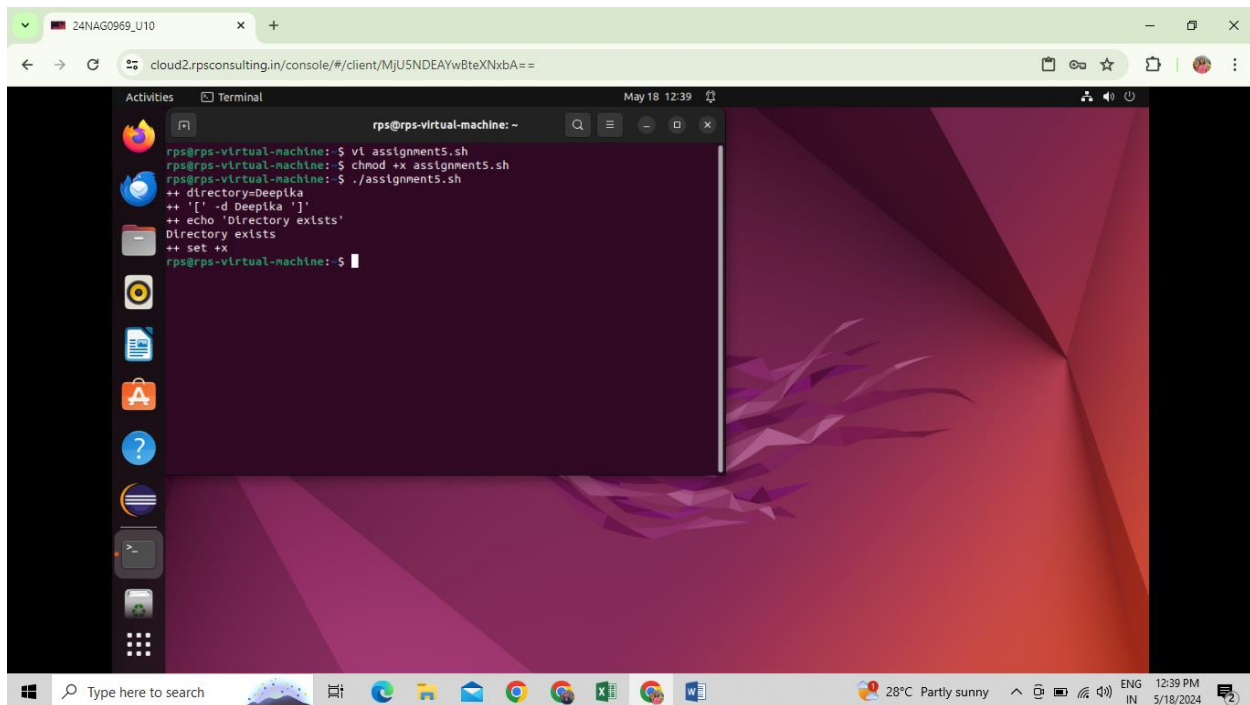


5] Modify the script to handle errors, such as the directory already existing or lacking permissions to create files.

Add a debugging mode that prints additional information when enabled.



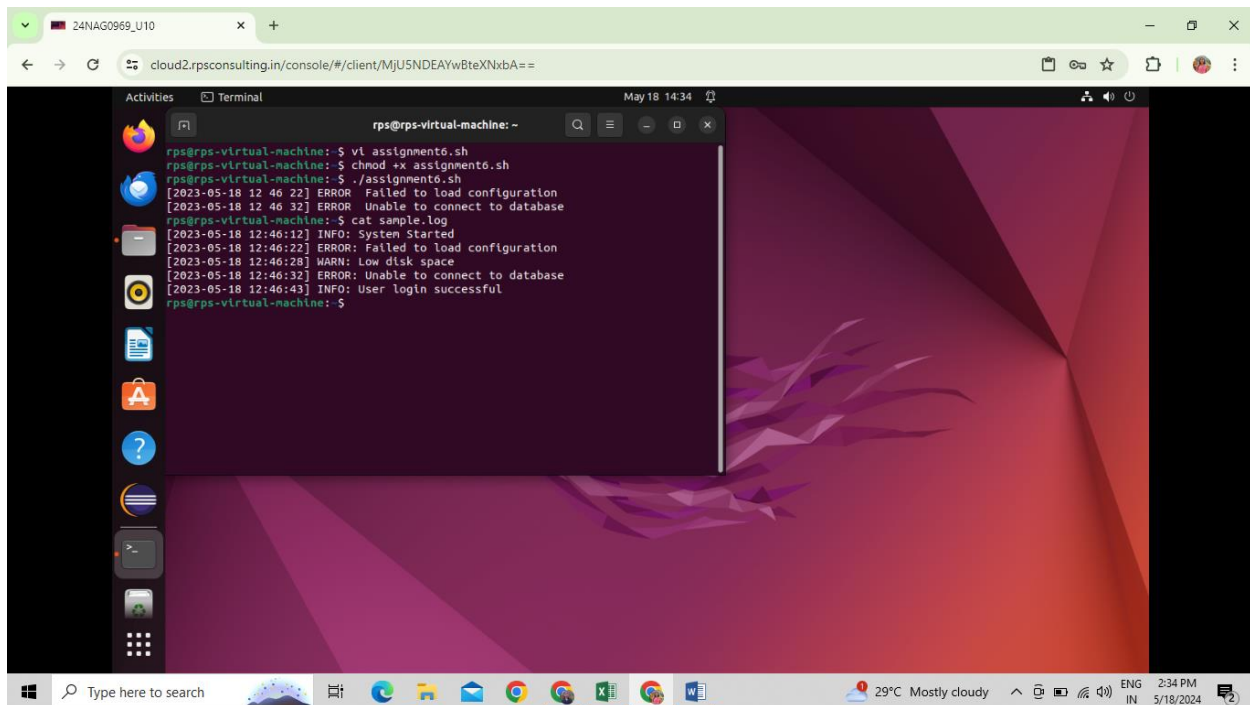
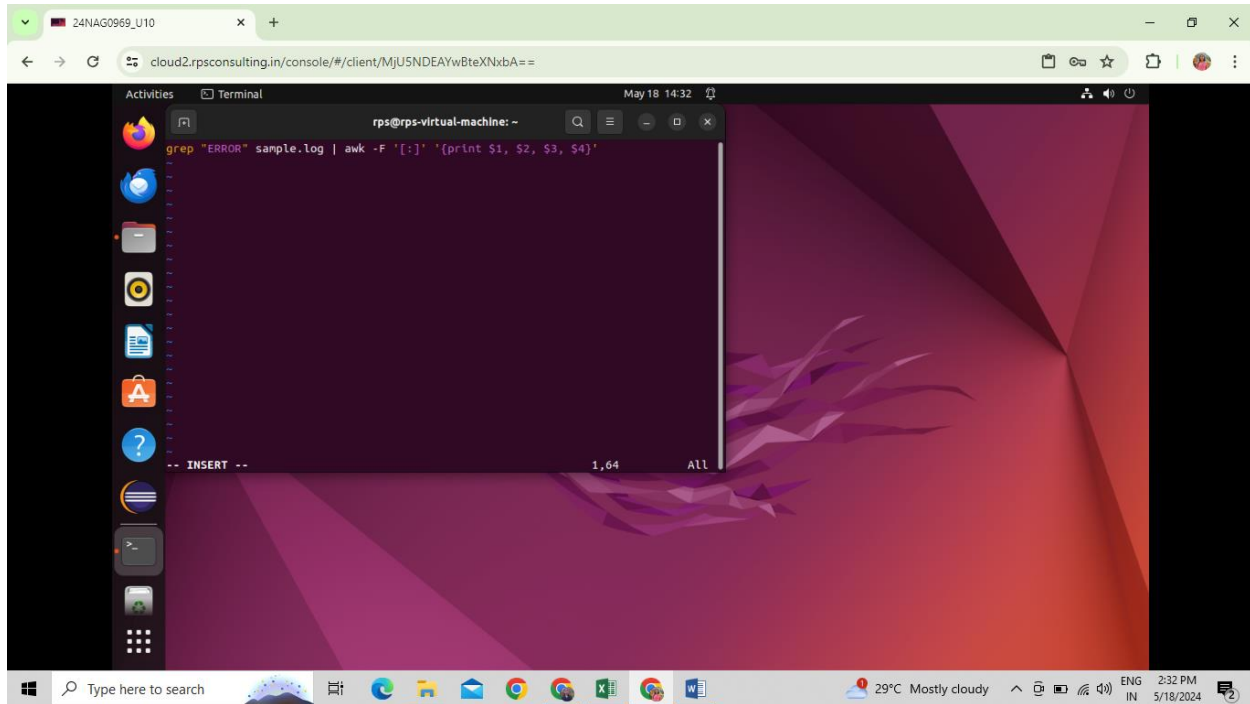
```
set -x
directory="Deepika"
if [ -d "$directory" ]
then
    echo "Directory exists"
else
    mkdir -p "$directory"
    echo "Directory created"
fi
set +x
```



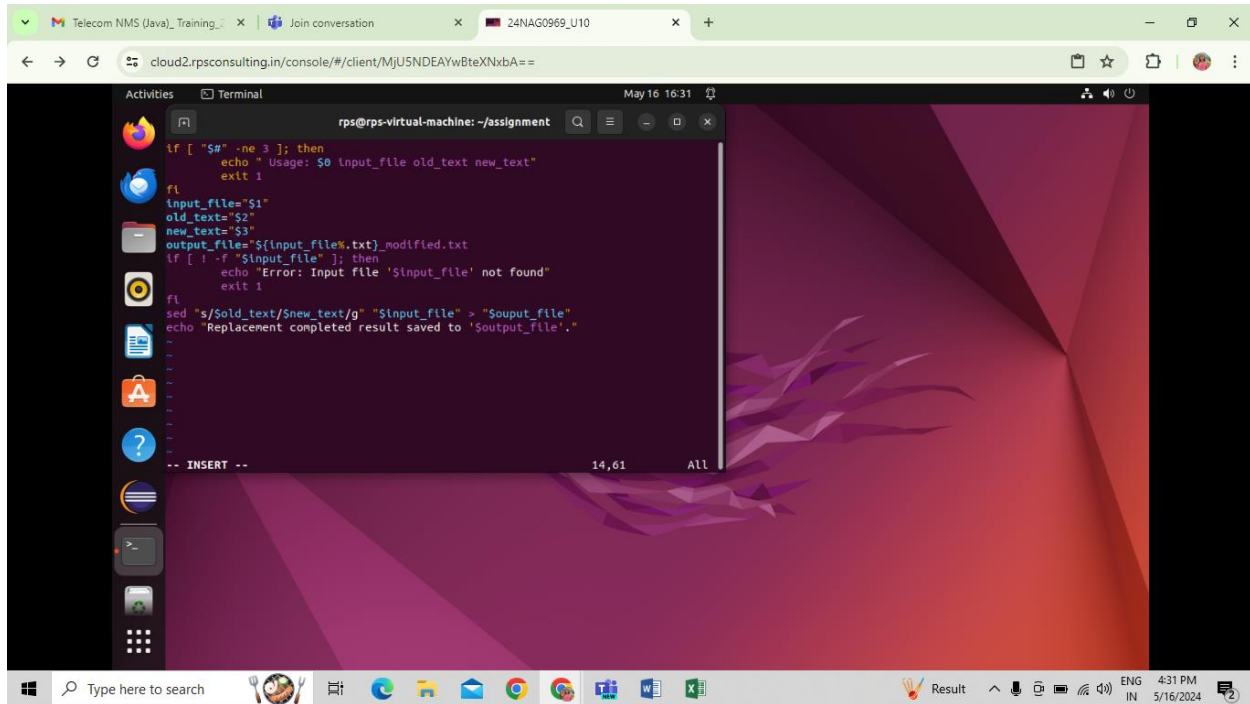
```
rps@rps-virtual-machine:~$ vi assignments5.sh
rps@rps-virtual-machine:~$ chmod +x assignments5.sh
rps@rps-virtual-machine:~$ ./assignments5.sh
++ directory=Deepika
++ '[' -d Deepika ']'
++ echo 'Directory exists'
Directory exists
++ set +x
rps@rps-virtual-machine:~$
```

6] Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line.

## Data Processing with sed



7] Create a script that takes a text file and replaces all occurrences of "old\_text" with "new\_text". Use sed to perform this operation and output the result to a new file.

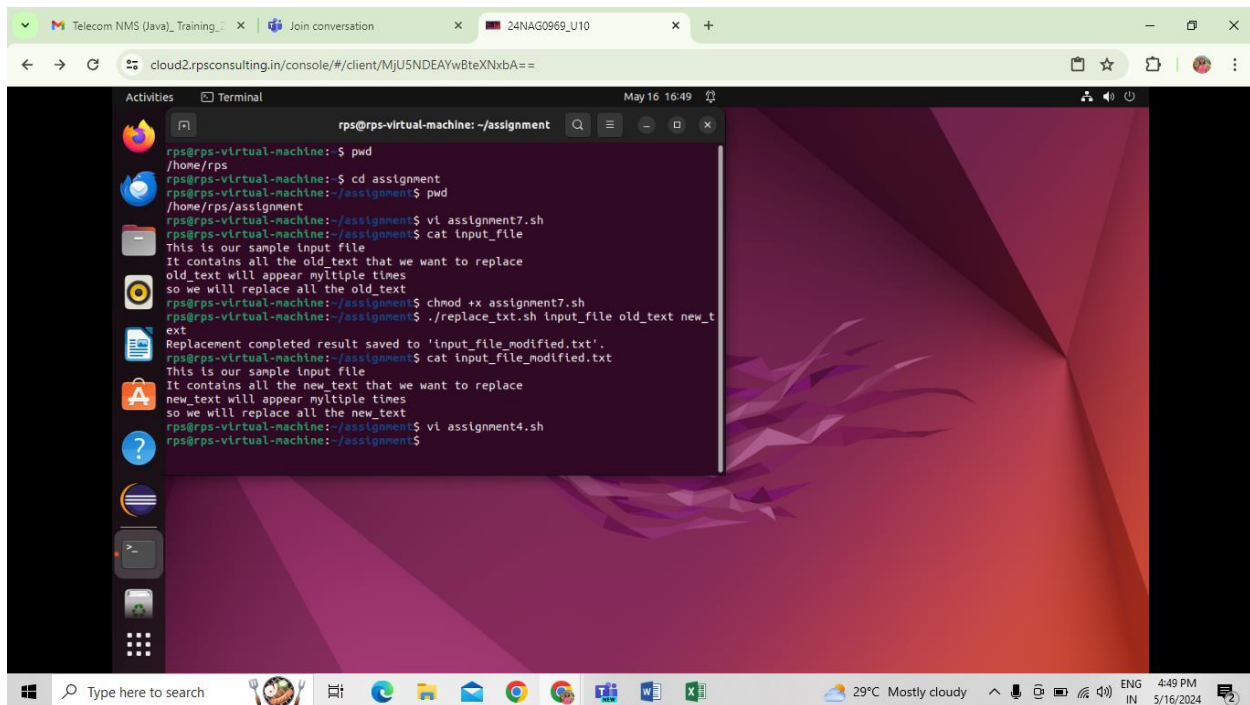


The screenshot shows a terminal window titled 'rps@rps-virtual-machine: ~/assignment'. The user has created a script named 'replace\_txt.sh' with the following content:

```
#!/bin/bash
if [ $# -ne 3 ]; then
    echo "Usage: $0 input_file old_text new_text"
    exit 1
fi
input_file="$1"
old_text="$2"
new_text="$3"
output_file="${input_file}.txt_modified.txt"
if [ ! -f "$input_file" ]; then
    echo "Error: Input file '$input_file' not found"
    exit 1
fi
sed "s/$old_text/$new_text/g" "$input_file" > "$output_file"
echo "Replacement completed result saved to '$output_file'."
```

The terminal also shows the file permissions and size of the script:

```
-- INSERT --
14,61 All
```



The screenshot shows the same terminal window with the following commands and output:

```
rps@rps-virtual-machine: $ pwd
/home/rps
rps@rps-virtual-machine: $ cd assignment
rps@rps-virtual-machine:~/assignment$ pwd
/home/rps/assignment
rps@rps-virtual-machine:~/assignment$ vi assignment7.sh
rps@rps-virtual-machine:~/assignment$ cat input_file
This is our sample input file
It contains all the old_text that we want to replace
old_text will appear myltiple times
so we will replace all the old_text
rps@rps-virtual-machine:~/assignment$ chmod +x assignment7.sh
rps@rps-virtual-machine:~/assignment$ ./replace_txt.sh input_file old_text new_t
ext
Replacement completed result saved to 'input_file_modified.txt'.
rps@rps-virtual-machine:~/assignment$ cat input_file_modified.txt
This is our sample input file
It contains all the new_text that we want to replace
new_text will appear myltiple times
so we will replace all the new_text
rps@rps-virtual-machine:~/assignment$ vi assignment4.sh
rps@rps-virtual-machine:~/assignment$
```