Task 4: Loop and Function Script in Bash

Objective:
To understand the use of functions and loops in Bash scripting. This task demonstrates how a for loop iterates over a list of files and how a function is used to count and display the number of lines in each file.

1. Concept Overview

Bash supports user-defined functions and looping constructs such as for loops. Functions help in code reusability, while loops allow repetitive execution of commands. The wc -l command is used to count the number of lines in a file.

2. Bash Script Code

```
#!/bin/bash

# Function to count lines in a file
count_lines() {
        local file=$1
        if [ -f "$file" ]; then
        lines=$(wc -l < "$file")
        echo "File: $file has $lines lines"
        else
        echo "File: $file does not exist"
        fi
}

# List of files
files=(file1.txt file2.txt file3.txt)

# For loop to iterate over files
for f in "${files[@]}"; do
        count_lines "$f"
done
```

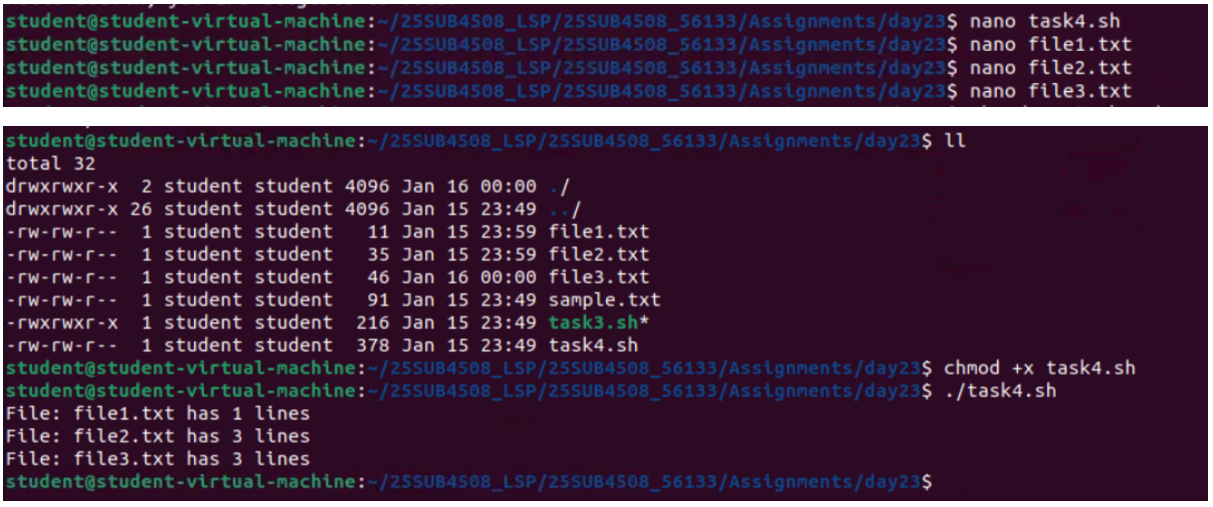
3. Script Explanation

1. A function count_lines() is defined to accept a filename as argument.
2. The function checks whether the file exists using -f.
3. wc -l is used to count the number of lines in the file.
4. An array of filenames is declared.
5. A for loop iterates over each file and calls the function.

4. Execution Steps

1. Create sample text files with some content.
2. Save the script as task4.sh
3. Give execute permission using chmod
4. Run the script from terminal

5. Sample Output



6. Observations

• Functions improve modularity of scripts.
• for loops simplify iteration over multiple items.
• wc -l is commonly used for line counting.

7. Conclusion

This task demonstrates how Bash functions and loops work together to process multiple files efficiently. Such scripts are widely used in system administration and automation tasks.