**Experiment : 5**

**Write a shell script to display the contents of a file between the given line numbers.**

#!/bin/bash  
  
# Check if the correct number of arguments is provided  
if [ "$#" -ne 3 ]; then  
    echo "Error: Incorrect number of arguments"  
    echo "Usage: $0 filename start\_line end\_line"  
    exit 1  
fi  
  
filename="$1"  
start\_line="$2"  
end\_line="$3"  
  
# Check if the file exists  
if [ ! -f "$filename" ]; then  
    echo "Error: File '$filename' does not exist"  
    exit 1  
fi  
  
# Check if start and end lines are valid numbers  
if ! [[ "$start\_line" =~ ^[0-9]+$ ]] || ! [[ "$end\_line" =~ ^[0-9]+$ ]]; then  
    echo "Error: Start and end line numbers must be positive integers"  
    exit 1  
fi  
  
# Check if start line is less than or equal to end line  
if [ "$start\_line" -gt "$end\_line" ]; then  
    echo "Error: Start line cannot be greater than end line"  
    exit 1  
fi  
  
line\_count=$(wc -l < "$1")  
  
echo " The total number of lines in $2: $line\_count"  
  
# Display the lines between start\_line and end\_line (inclusive)  
sed -n "${start\_line},${end\_line}p" "$filename"

**EXPERIMENT 6**

**Write a shell script the deletes all lines containing a specified word among the files.**

#!/bin/bash  
  
# Check if at least two arguments are provided  
if [ $# -lt 2 ]; then  
    echo "Usage: $0 <word> <file1> [file2 ...]"  
    exit 1  
fi  
  
# The first argument is the word to search for  
word="$1"  
shift  
  
# Loop through all the files provided as arguments  
for file in "$@"  
do  
    # Check if the file exists  
    if [ ! -f "$file" ]; then  
        echo "File not found: $file"  
        continue  
    fi  
  
    # Use sed to delete lines containing the word and save to a temporary file  
    sed "/$word/d" "$file" > "$file.tmp"  
  
    # Replace the original file with the modified one  
    mv "$file.tmp" "$file"  
  
    echo "Processed $file: Removed lines containing '$word'"  
done  
  
echo "All files processed."