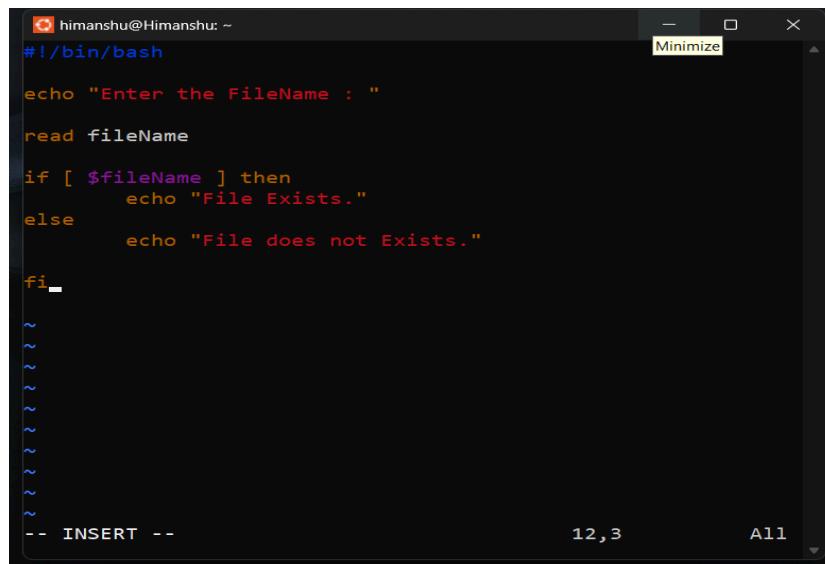


Himanshu Kamane  
[kamanehimanshu76@gmail.com](mailto:kamanehimanshu76@gmail.com)

Que 1-Assignment 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".

Answer=>

1. Shell Script code:



```
himanshu@Himanshu: ~
#!/bin/bash

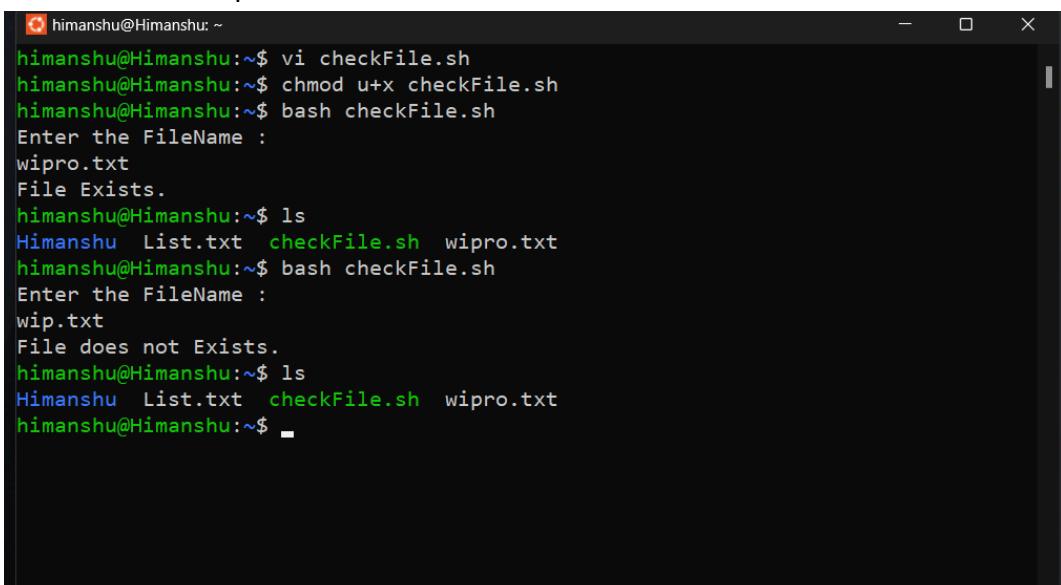
echo "Enter the FileName : "

read fileName

if [ $fileName ] then
    echo "File Exists."
else
    echo "File does not Exists."

fi
```

2. Execution and output:

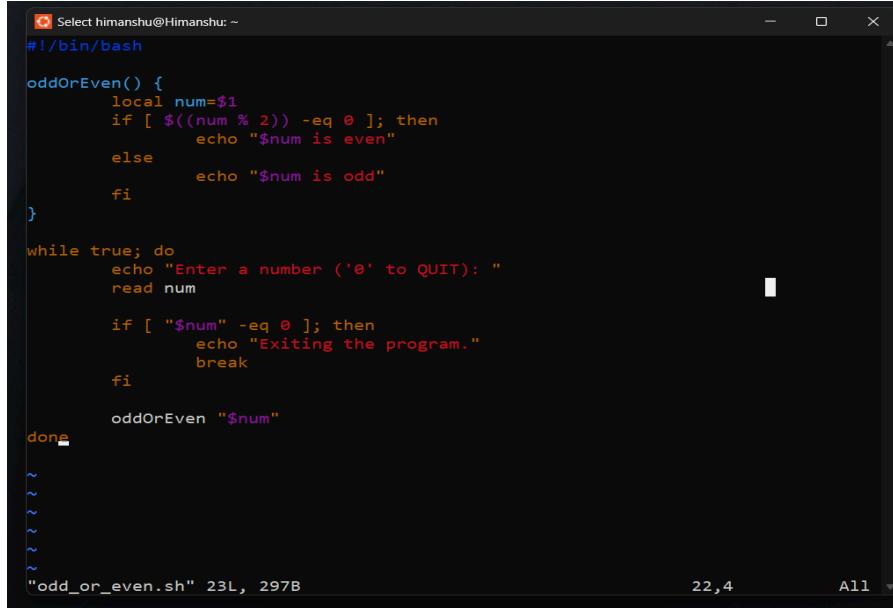


```
himanshu@Himanshu:~$ vi checkFile.sh
himanshu@Himanshu:~$ chmod u+x checkFile.sh
himanshu@Himanshu:~$ bash checkFile.sh
Enter the FileName :
wipro.txt
File Exists.
himanshu@Himanshu:~$ ls
Himanshu List.txt  checkFile.sh  wipro.txt
himanshu@Himanshu:~$ bash checkFile.sh
Enter the FileName :
wip.txt
File does not Exists.
himanshu@Himanshu:~$ ls
Himanshu List.txt  checkFile.sh  wipro.txt
himanshu@Himanshu:~$
```

Que2-Assignment 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.

Answer=>

1. Shell Script code:



```
#!/bin/bash

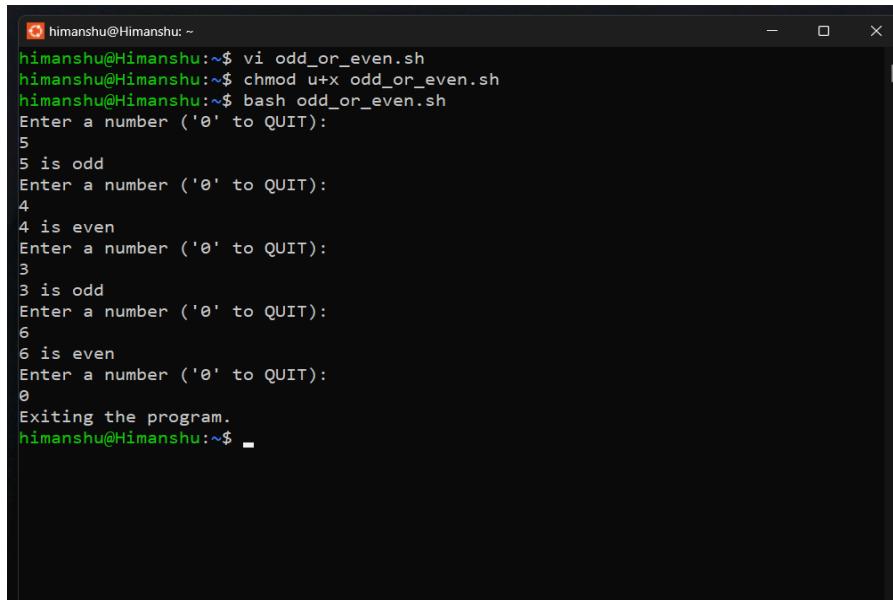
oddOrEven() {
    local num=$1
    if [ $((num % 2)) -eq 0 ]; then
        echo "$num is even"
    else
        echo "$num is odd"
    fi
}

while true; do
    echo "Enter a number ('0' to QUIT): "
    read num

    if [ "$num" -eq 0 ]; then
        echo "Exiting the program."
        break
    fi

    oddOrEven "$num"
done
```

2. Execution and output:



```
himanshu@Himanshu:~$ vi odd_or_even.sh
himanshu@Himanshu:~$ chmod u+x odd_or_even.sh
himanshu@Himanshu:~$ bash odd_or_even.sh
Enter a number ('0' to QUIT):
5
5 is odd
Enter a number ('0' to QUIT):
4
4 is even
Enter a number ('0' to QUIT):
3
3 is odd
Enter a number ('0' to QUIT):
6
6 is even
Enter a number ('0' to QUIT):
0
Exiting the program.
himanshu@Himanshu:~$
```

Que 3 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

Answer=>

## 1. Shell Script code:

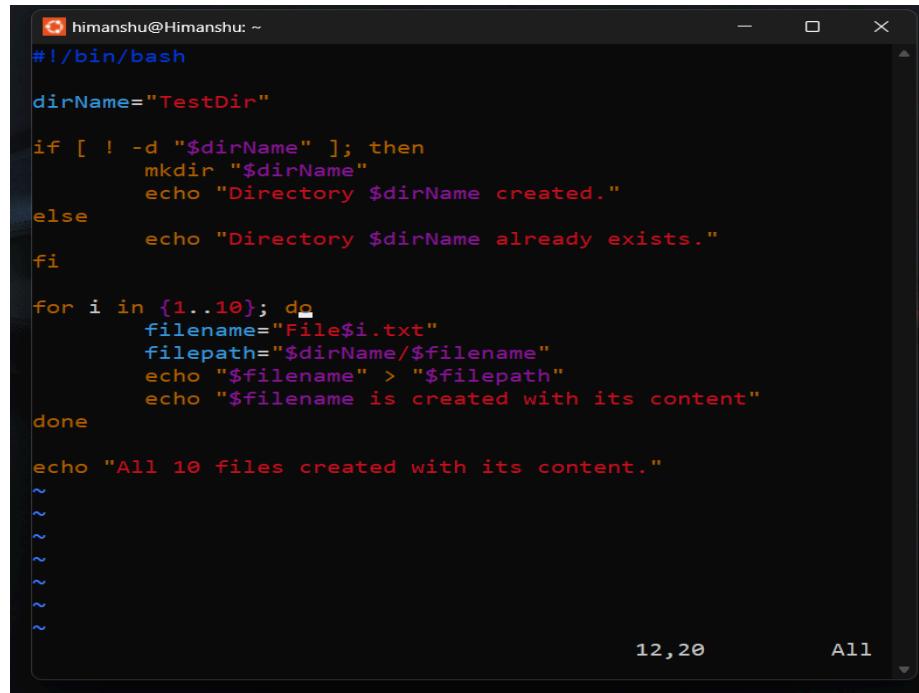
## 2. Execution and output:

```
himanshu@Himanshu:~$ vi linesInFile.sh
himanshu@Himanshu:~$ chmod u+x linesInFile.sh
himanshu@Himanshu:~$ bash linesInFile.sh
Enter the filename:
wipro.txt
The file wipro.txt has 1 lines.
himanshu@Himanshu:~$ cat wipro.txt
Hello i am wipro
himanshu@Himanshu:~$ bash linesInFile.sh
Enter the filename:
List.txt
The file List.txt has 3 lines.
himanshu@Himanshu:~$ cat List.txt
List1
List2
List3
himanshu@Himanshu:~$
```

Que4 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").

Answer:

1. Shell Script code:



```
himanshu@Himanshu: ~
#!/bin/bash

dirName="TestDir"

if [ ! -d "$dirName" ]; then
    mkdir "$dirName"
    echo "Directory $dirName created."
else
    echo "Directory $dirName already exists."
fi

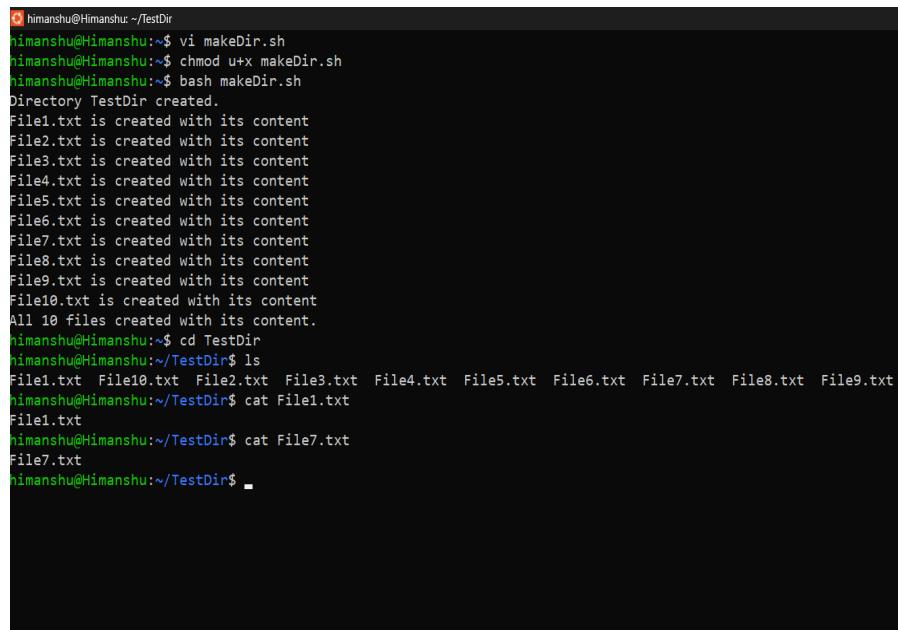
for i in {1..10}; do
    filename="File$i.txt"
    filepath="$dirName/$filename"
    echo "$filename" > "$filepath"
    echo "$filename is created with its content"
done

echo "All 10 files created with its content."
~
```

12,20

All

2. Execution and Output:

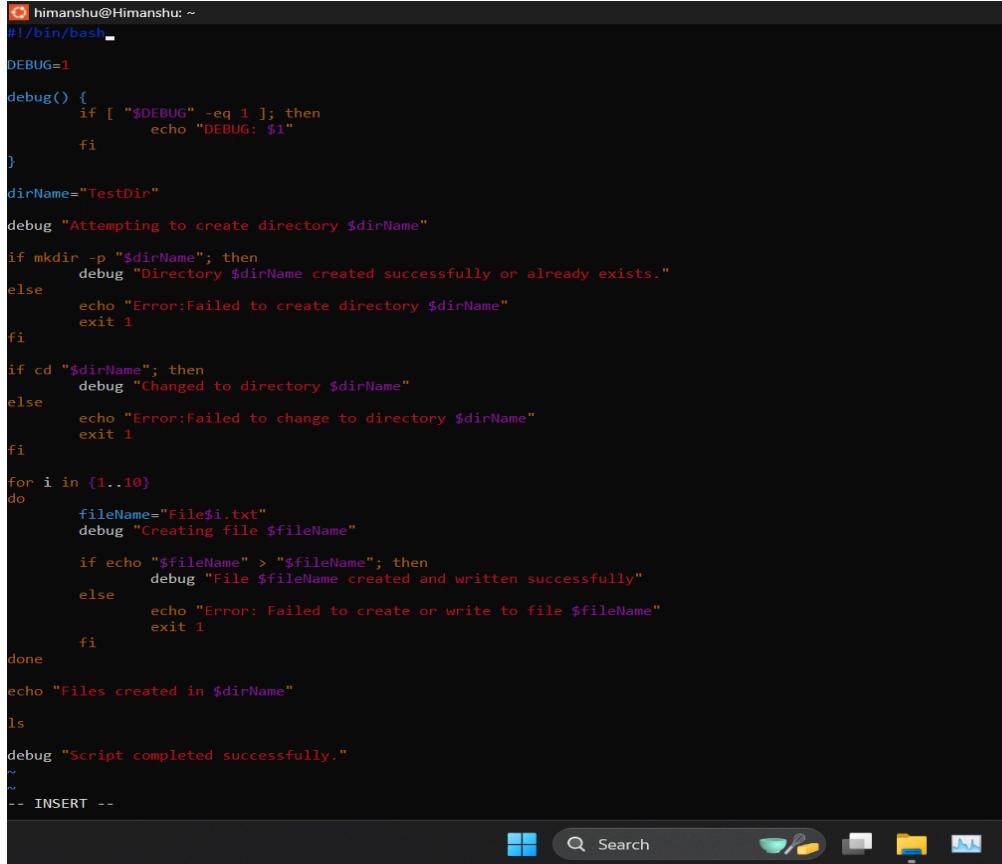


```
himanshu@Himanshu:~/TestDir
himanshu@Himanshu:~$ vi makeDir.sh
himanshu@Himanshu:~$ chmod u+x makeDir.sh
himanshu@Himanshu:~$ bash makeDir.sh
Directory TestDir created.
File1.txt is created with its content
File2.txt is created with its content
File3.txt is created with its content
File4.txt is created with its content
File5.txt is created with its content
File6.txt is created with its content
File7.txt is created with its content
File8.txt is created with its content
File9.txt is created with its content
File10.txt is created with its content
All 10 files created with its content.
himanshu@Himanshu:~$ cd TestDir
himanshu@Himanshu:~/TestDir$ ls
File1.txt File10.txt File2.txt File3.txt File4.txt File5.txt File6.txt File7.txt File8.txt File9.txt
himanshu@Himanshu:~/TestDir$ cat File1.txt
File1.txt
himanshu@Himanshu:~/TestDir$ cat File7.txt
File7.txt
himanshu@Himanshu:~/TestDir$
```

Que 5 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.

Answer=>

1. Shell Script code:



```
himanshu@Himanshu: ~
#!/bin/bash

DEBUG=1

debug() {
    if [ "$DEBUG" -eq 1 ]; then
        echo "DEBUG: $1"
    fi
}

dirName="TestDir"

debug "Attempting to create directory $dirName"

if mkdir -p "$dirName"; then
    debug "Directory $dirName created successfully or already exists."
else
    echo "Error:Failed to create directory $dirName"
    exit 1
fi

if cd "$dirName"; then
    debug "Changed to directory $dirName"
else
    echo "Error:Failed to change to directory $dirName"
    exit 1
fi

for i in {1..10}
do
    fileName="File$i.txt"
    debug "Creating file $fileName"

    if echo "$fileName" > "$fileName"; then
        debug "File $fileName created and written successfully"
    else
        echo "Error: Failed to create or write to file $fileName"
        exit 1
    fi
done

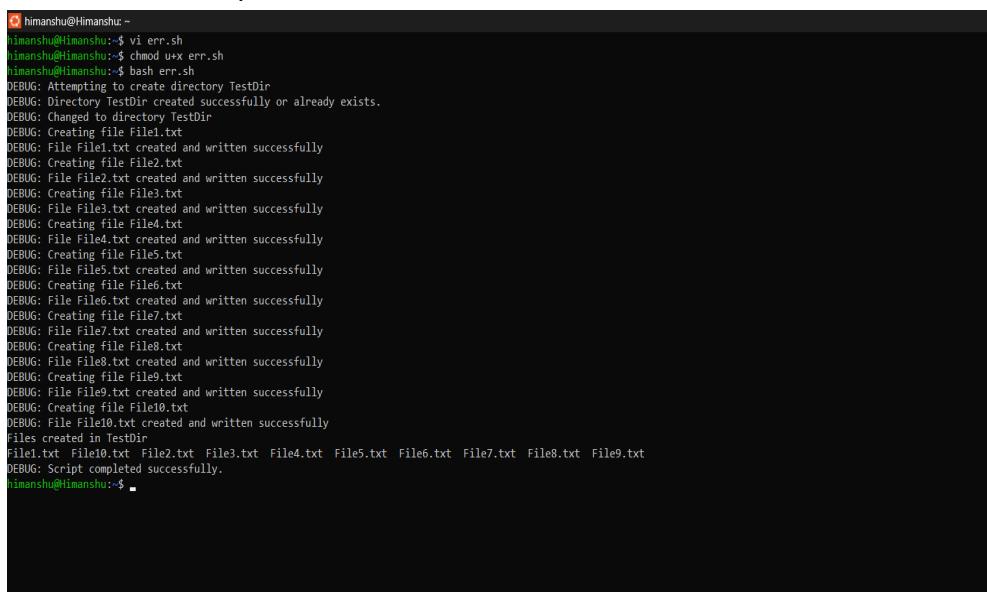
echo "Files created in $dirName"

ls

debug "Script completed successfully."
~
```

-- INSERT --

2. Execution and output:



```
himanshu@Himanshu: ~
himanshu@Himanshu:~$ vi err.sh
himanshu@Himanshu:~$ chmod uwx err.sh
himanshu@Himanshu:~$ bash err.sh
DEBUG: Attempting to create directory TestDir
DEBUG: Directory TestDir created successfully or already exists.
DEBUG: Changed to directory TestDir
DEBUG: Creating file File1.txt
DEBUG: File File1.txt created and written successfully
DEBUG: Creating file File2.txt
DEBUG: File File2.txt created and written successfully
DEBUG: Creating file File3.txt
DEBUG: File File3.txt created and written successfully
DEBUG: Creating file File4.txt
DEBUG: File File4.txt created and written successfully
DEBUG: Creating file File5.txt
DEBUG: File File5.txt created and written successfully
DEBUG: Creating file File6.txt
DEBUG: File File6.txt created and written successfully
DEBUG: Creating file File7.txt
DEBUG: File File7.txt created and written successfully
DEBUG: Creating file File8.txt
DEBUG: File File8.txt created and written successfully
DEBUG: Creating file File9.txt
DEBUG: File File9.txt created and written successfully
DEBUG: Creating file File10.txt
DEBUG: File File10.txt created and written successfully
Files created in TestDir
File1.txt File10.txt File2.txt File3.txt File4.txt File5.txt File6.txt File7.txt File8.txt File9.txt
DEBUG: Script completed successfully.
himanshu@Himanshu:~$
```

Que 6 Assignment 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.

Answer=>

1. Shell Script code:

```
>Select himanshu@Himanshu: ~
#!/bin/bash

logFile="Sample.log"

if [ ! -f "$logFile" ]; then
    echo "Log file $logFile does not exist."
    exit 1
fi

grep "ERROR" "$logFile" | awk '{print $1, $2, substr($0, index($0,$4))}'
```

2. Sample.log file, execution and output:

```
>Select himanshu@Himanshu: ~$ cat Sample.log
himanshu@Himanshu:~$ cat Sample.log
2024-05-18 10:22:35 INFO Starting the application
2024-05-18 10:22:37 ERROR Failed to load configuration file
2024-05-18 10:22:38 INFO Configuration loaded successfully
2024-05-18 10:22:40 WARN Disk space low
2024-05-18 10:22:42 ERROR Disk space is low
2024-05-18 10:22:45 INFO User login successful
2024-05-18 10:22:50 ERROR Unexpected token in JSON at position 23
2024-05-18 10:22:55 INFO Application terminated
himanshu@Himanshu:~$ vi logfile.sh
himanshu@Himanshu:~$ chmod u+x logfile.sh
himanshu@Himanshu:~$ bash logfile.sh
2024-05-18 10:22:37 Failed to load configuration file
2024-05-18 10:22:42 Disk space is low
2024-05-18 10:22:50 Unexpected token in JSON at position 23
himanshu@Himanshu:~$
```

Que 7 Assignment 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.

Answer=>

1. Shell script code:

```
himanshu@Himanshu: ~
#!/bin/bash

if [ "$#" -ne 3 ]; then
    echo "Usage: $0 <inputFile> <oldText> <newText>"
    exit 1
fi

inputFile="$1"
oldText="$2"
newText="$3"

outputFile="Result.txt"

if [ ! -f "$inputFile" ]; then
    echo "Error: Input file '$inputFile' does not exist."
    exit
fi

sed "s/$oldText/$newText/g""$inputFile">"$outputFile"

echo "All occurrences of $oldText have been replaced with $newText in $outputFile."
~
```

2. Sample.txt file, execution and output in Result.txt file:

```
himanshu@Himanshu: ~
himanshu@Himanshu:~$ vi newText.sh
himanshu@Himanshu:~$ chmod u+x newText.sh
himanshu@Himanshu:~$ cat sample.txt
This is the demo file
demo file is used to write rough text
demo file can be used for making note
himanshu@Himanshu:~$ ./newText.sh sample.txt demo Parctice
All occurrences of demo have been replaced with Parctice in Result.txt.
himanshu@Himanshu:~$ ls
Himanshu  Result.txt  TestDir      err.sh      logfile.sh  newText.sh   sample.txt
List.txt  Sample.log  checkFile.sh  linesInFile.sh  makeDir.sh  odd_or_even.sh  wipro.txt
himanshu@Himanshu:~$ cat Result.sh
cat: Result.sh: No such file or directory
himanshu@Himanshu:~$ cat Result.txt
This is the Parctice file
Parctice file is used to write rough text
Parctice file can be used for making note
himanshu@Himanshu:~$
```