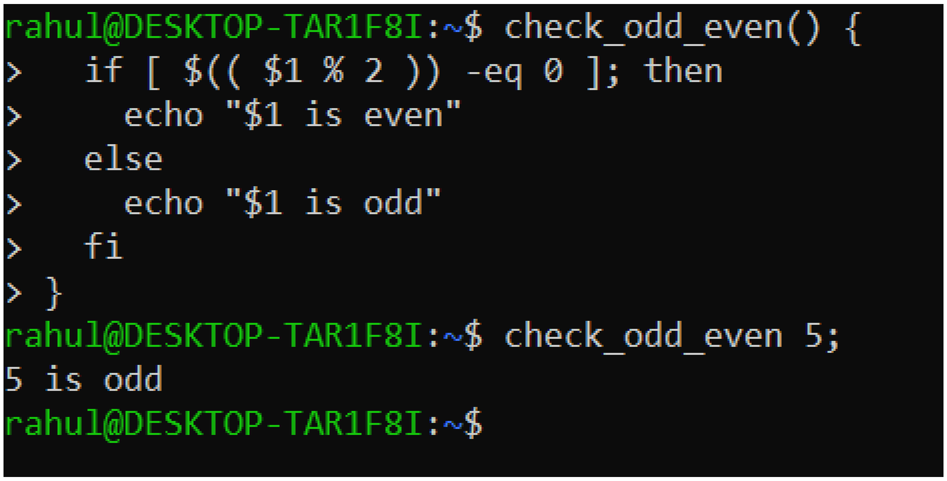
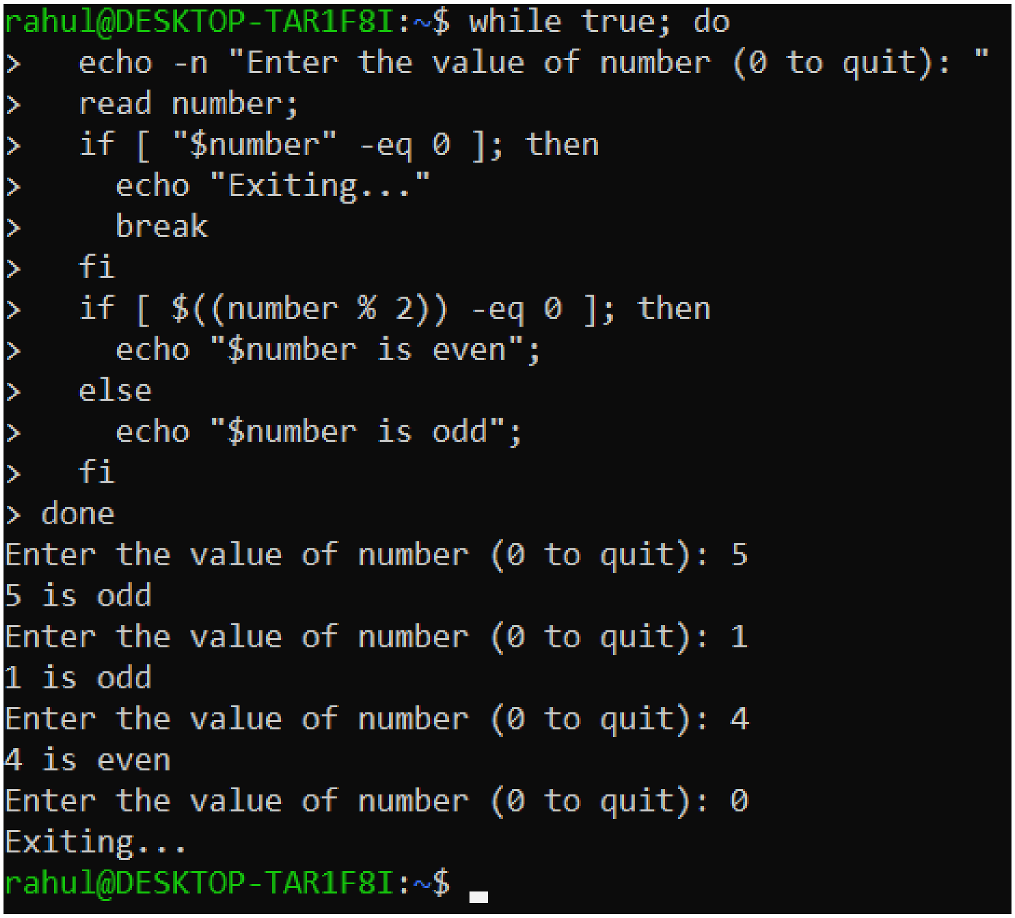
❖ **Assignment 1:** Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".



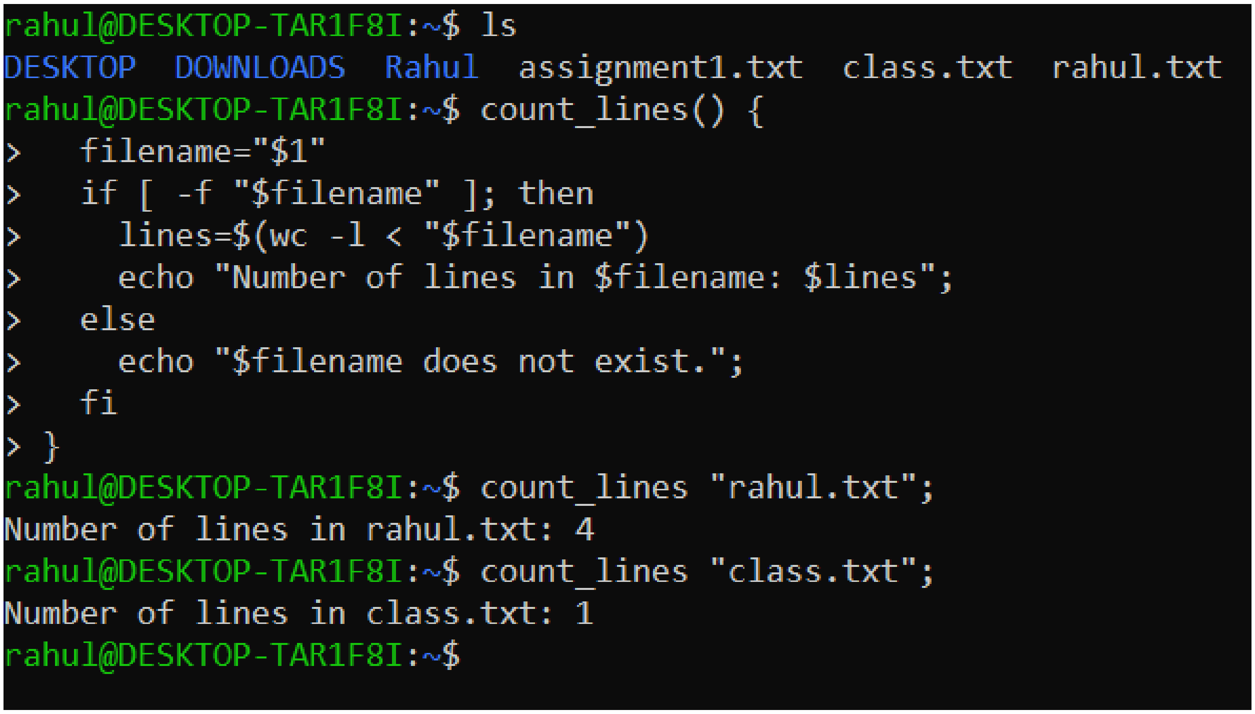
```
rahul@DESKTOP-TAR1F8I:~$ check_odd_even() {
>   if [ $(( $1 % 2 )) -eq 0 ]; then
>     echo "$1 is even"
>   else
>     echo "$1 is odd"
>   fi
> }
rahul@DESKTOP-TAR1F8I:~$ check_odd_even 5;
5 is odd
rahul@DESKTOP-TAR1F8I:~$
```

❖ **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.



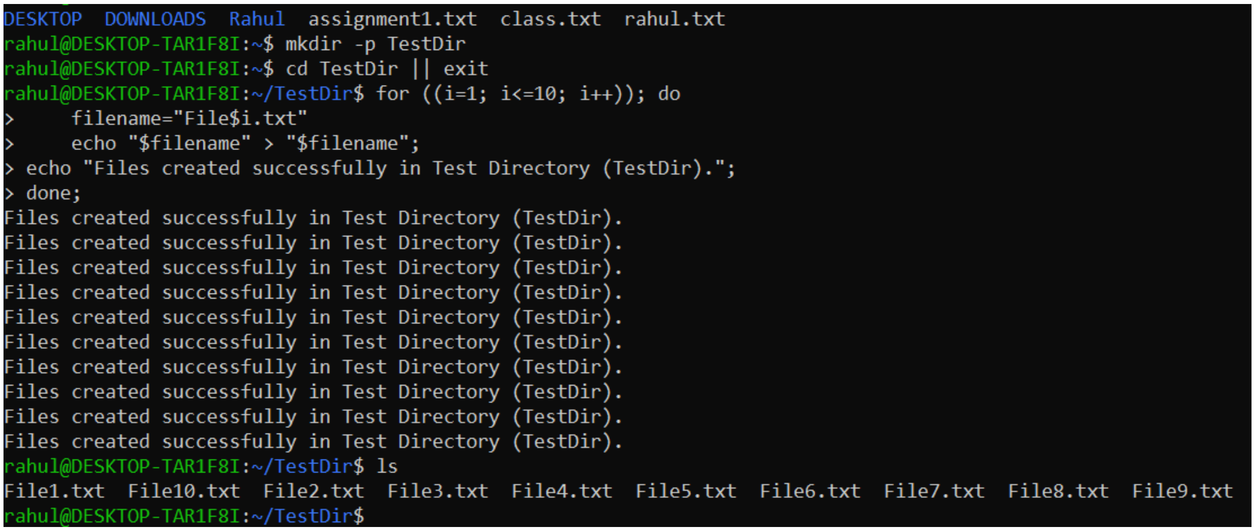
```
rahul@DESKTOP-TAR1F8I:~$ while true; do
>   echo -n "Enter the value of number (0 to quit): "
>   read number;
>   if [ "$number" -eq 0 ]; then
>     echo "Exiting..."
>     break
>   fi
>   if [ $((number % 2)) -eq 0 ]; then
>     echo "$number is even";
>   else
>     echo "$number is odd";
>   fi
> done
Enter the value of number (0 to quit): 5
5 is odd
Enter the value of number (0 to quit): 1
1 is odd
Enter the value of number (0 to quit): 4
4 is even
Enter the value of number (0 to quit): 0
Exiting...
rahul@DESKTOP-TAR1F8I:~$
```

- **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.



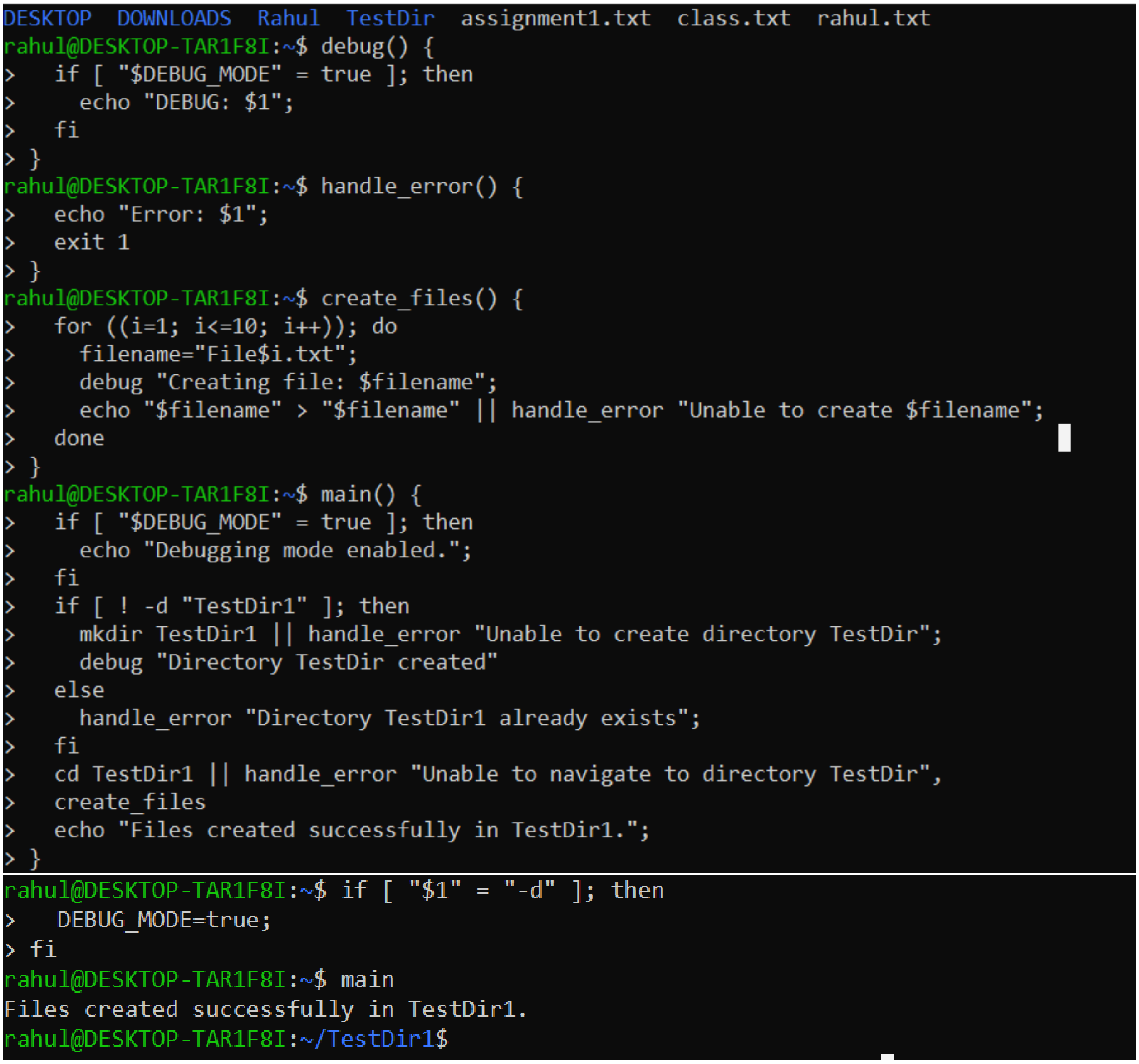
```
rahul@DESKTOP-TAR1F8I:~$ ls
DESKTOP  DOWNLOADS  Rahul  assignment1.txt  class.txt  rahul.txt
rahul@DESKTOP-TAR1F8I:~$ count_lines() {
>   filename="$1"
>   if [ -f "$filename" ]; then
>     lines=$(wc -l < "$filename")
>     echo "Number of lines in $filename: $lines";
>   else
>     echo "$filename does not exist.";
>   fi
> }
rahul@DESKTOP-TAR1F8I:~$ count_lines "rahul.txt";
Number of lines in rahul.txt: 4
rahul@DESKTOP-TAR1F8I:~$ count_lines "class.txt";
Number of lines in class.txt: 1
rahul@DESKTOP-TAR1F8I:~$
```

- **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").



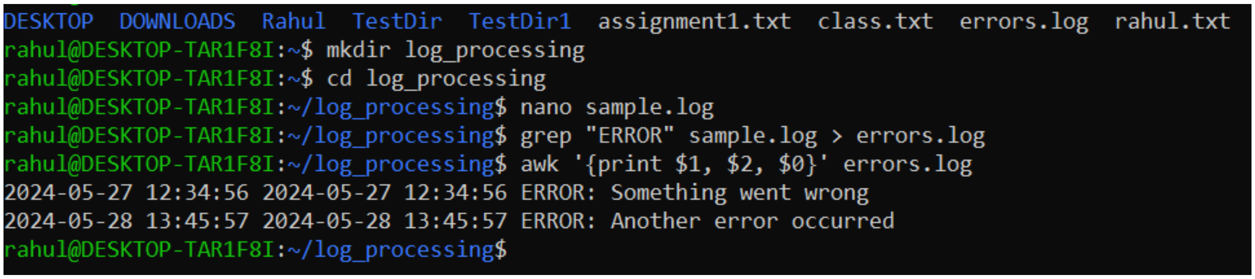
```
DESKTOP  DOWNLOADS  Rahul  assignment1.txt  class.txt  rahul.txt
rahul@DESKTOP-TAR1F8I:~$ mkdir -p TestDir
rahul@DESKTOP-TAR1F8I:~$ cd TestDir || exit
rahul@DESKTOP-TAR1F8I:~/TestDir$ for ((i=1; i<=10; i++)); do
>     filename="File$i.txt"
>     echo "$filename" > "$filename";
> echo "Files created successfully in Test Directory (TestDir).";
> done;
Files created successfully in Test Directory (TestDir).
Files created successfully in Test Directory (TestDir).
Files created successfully in Test Directory (TestDir).
Files created successfully in Test Directory (TestDir).
Files created successfully in Test Directory (TestDir).
Files created successfully in Test Directory (TestDir).
Files created successfully in Test Directory (TestDir).
Files created successfully in Test Directory (TestDir).
Files created successfully in Test Directory (TestDir).
Files created successfully in Test Directory (TestDir).
rahul@DESKTOP-TAR1F8I:~/TestDir$ ls
File1.txt  File10.txt  File2.txt  File3.txt  File4.txt  File5.txt  File6.txt  File7.txt  File8.txt  File9.txt
rahul@DESKTOP-TAR1F8I:~/TestDir$
```

- **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.



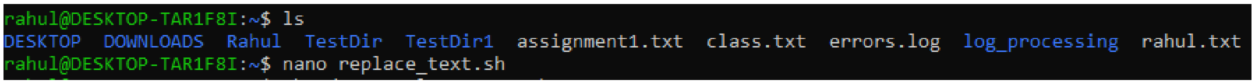
```
DESKTOP  DOWNLOADS  Rahul  TestDir  assignment1.txt  class.txt  rahul.txt
rahul@DESKTOP-TAR1F8I:~$ debug() {
>   if [ "$DEBUG_MODE" = true ]; then
>     echo "DEBUG: $1";
>   fi
> }
rahul@DESKTOP-TAR1F8I:~$ handle_error() {
>   echo "Error: $1";
>   exit 1
> }
rahul@DESKTOP-TAR1F8I:~$ create_files() {
>   for ((i=1; i<=10; i++)); do
>     filename="File$i.txt";
>     debug "Creating file: $filename";
>     echo "$filename" > "$filename" || handle_error "Unable to create $filename";
>   done
> }
rahul@DESKTOP-TAR1F8I:~$ main() {
>   if [ "$DEBUG_MODE" = true ]; then
>     echo "Debugging mode enabled.";
>   fi
>   if [ ! -d "TestDir1" ]; then
>     mkdir TestDir1 || handle_error "Unable to create directory TestDir";
>     debug "Directory TestDir created"
>   else
>     handle_error "Directory TestDir1 already exists";
>   fi
>   cd TestDir1 || handle_error "Unable to navigate to directory TestDir",
>   create_files
>   echo "Files created successfully in TestDir1.";
> }
rahul@DESKTOP-TAR1F8I:~$ if [ "$1" = "-d" ]; then
>   DEBUG_MODE=true;
> fi
rahul@DESKTOP-TAR1F8I:~$ main
Files created successfully in TestDir1.
rahul@DESKTOP-TAR1F8I:~/TestDir1$
```

- **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. Data Processing with sed

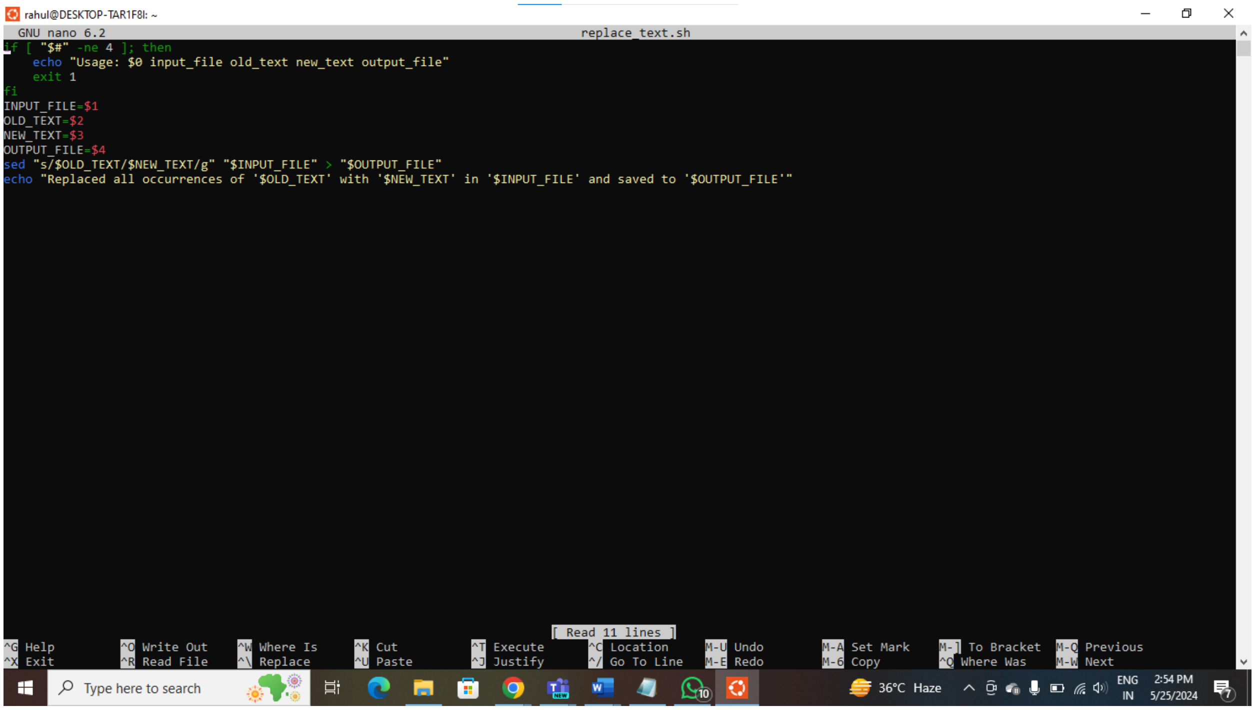


```
DESKTOP  DOWNLOADS  Rahul  TestDir  TestDir1  assignment1.txt  class.txt  errors.log  rahul.txt
rahul@DESKTOP-TAR1F8I:~$ mkdir log_processing
rahul@DESKTOP-TAR1F8I:~$ cd log_processing
rahul@DESKTOP-TAR1F8I:~/log_processing$ nano sample.log
rahul@DESKTOP-TAR1F8I:~/log_processing$ grep "ERROR" sample.log > errors.log
rahul@DESKTOP-TAR1F8I:~/log_processing$ awk '{print $1, $2, $0}' errors.log
2024-05-27 12:34:56 2024-05-27 12:34:56 ERROR: Something went wrong
2024-05-28 13:45:57 2024-05-28 13:45:57 ERROR: Another error occurred
rahul@DESKTOP-TAR1F8I:~/log_processing$
```

- **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.

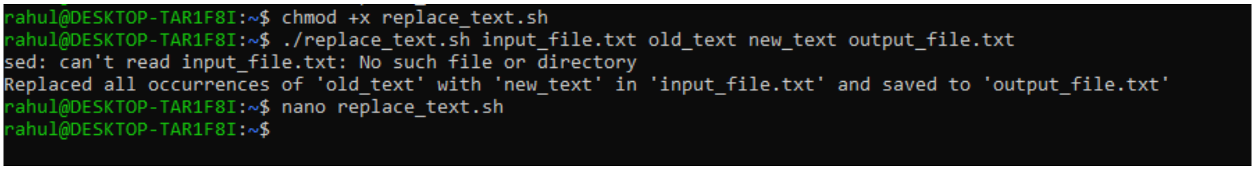


```
rahul@DESKTOP-TAR1F8I:~$ ls
DESKTOP  DOWNLOADS  Rahul  TestDir  TestDir1  assignment1.txt  class.txt  errors.log  log_processing  rahul.txt
rahul@DESKTOP-TAR1F8I:~$ nano replace_text.sh
```



```
if [ "$#" -ne 4 ]; then
    echo "Usage: $0 input_file old_text new_text output_file"
    exit 1
fi
INPUT_FILE=$1
OLD_TEXT=$2
NEW_TEXT=$3
OUTPUT_FILE=$4
sed "s/$OLD_TEXT/$NEW_TEXT/g" "$INPUT_FILE" > "$OUTPUT_FILE"
echo "Replaced all occurrences of '$OLD_TEXT' with '$NEW_TEXT' in '$INPUT_FILE' and saved to '$OUTPUT_FILE'"
```

PRESS CTRL + O , THEN PRESS ENTER, THEN PRESS CTRL + X



```
rahul@DESKTOP-TAR1F8I:~$ chmod +x replace_text.sh
rahul@DESKTOP-TAR1F8I:~$ ./replace_text.sh input_file.txt old_text new_text output_file.txt
sed: can't read input_file.txt: No such file or directory
Replaced all occurrences of 'old_text' with 'new_text' in 'input_file.txt' and saved to 'output_file.txt'
rahul@DESKTOP-TAR1F8I:~$ nano replace_text.sh
rahul@DESKTOP-TAR1F8I:~$
```