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".

Logic(code):

```
$ cat fileCheck.sh
file="sum.sh"
if [ "$file" ]; then
 echo "File exisits"
else
 echo "File not found"
 fi
```

Output:-

```
$ vim fileCheck.sh
Administrator@DESKTOP-TIC5DM4 MINGW64 ~/git_demo2 (master)
$ ./fileCheck.sh
File exisits
```

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.

LOGIC(CODE):

```
$ cat even_odd.sh
while true; do
 read -p "Enter a number (0 to exist): " number
 if [ "$number" -eq 0 ]; then
 break
 fi
 if [ $(($number % 2)) -eq 0 ]; then
 echo "$number is even"
```

```
 else
 echo "$number is odd"
 fi
done
```

Output:-

```
$ nano even_odd.sh
$ ./even_odd.sh
Enter a number (0 to exist): 4
4 is even
Enter a number (0 to exist): 5
5 is odd
Enter a number (0 to exist):
```

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.

Logic (code):

```
$ cat lines.sh
lines=$(wc -l < sum.sh)
echo "Number of lines present in sum.sh is $lines"
```

OUTPUT:-

```
$ ./lines.sh
Number of lines present in sum.sh is 6
```

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").

LOGIC(CODE):-

```
$ cat g1.sh
mkdir -p TestDir
cd TestDir
for i in {1..10}
do
 echo "Files{i}.txt"> "File${i}.txt"
done
```

OUTPUT:-

```
Administrator@DESKTOP-TIC5DM4 MINGW64 ~/git_demo2 (master)
$ chmod +x g1.sh
Administrator@DESKTOP-TIC5DM4 MINGW64 ~/git_demo2 (master)
$ ./g1.sh
Files created in TestDir:
total 10
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File1.txt
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File10.txt
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File2.txt
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File3.txt
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File4.txt
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File5.txt
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File6.txt
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File7.txt
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File8.txt
-rw-r--r-- 1 Administrator 197121 13 Jun 27 11:07 File9.txt
```

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.

LOGIC(CODE):-

```
$#!/bin/bash

$ debug mode="off"

Administrator BDESKTOP-TICSDM4 MINGW64/(master)

$ debug print() [

if [ "$debug_mode" echo "DEBUG: 50" "on" ]; then

Administrator@DESKTOP-TIC5DM4 MINGW64/ (master)

$ temp_dir="temp_files"

mkdir -p "Stemp_dir" 2>/dev/null

Aduintstrator DESKTOP-TIC5DM4 MINGW64/ (master)

$ if [ $?-ne 0 ]; then

echo "Error: Could not create temporary directory 'Stemp_dir"."

exit 1

fi

debug print "Temporary directory created: Stemp_dir"

Administrator@DESKTOP-TIC5DM4 MINGW64/ (master)

$ text="This is some text with old_text in it. Here's another occurrence of old text."

Administrator DESKTOP-TIC50M4 MINGW64/(master)

$ temp file=$(mktemp-p "Stemp_dir" temp XXXXXX.txt)

Administrator DESKTOP-TIC SCM4 MINGW64/(master)

$if [ $?-ne 0 ]; then echo "Error: Could not create temporary file."

exit 1

fi

debug print "Temporary file created: Stemp_file"

Administrator DESKTOP-TIC5DM4 MINGW64/(master)

S echo "Stext" > "Stemp_file"

Administrator DESKTOP-TIC50M4 MINGW64 / (master)

$ sed "s/Sold_text/$new_text/g" "$temp_file" > "modified.txt"

Administrator DESKTOP-TIC5DM4 MINGW64 / (master)
```

$ if [$?-ne 0]; then

echo "Error: sed command failed during replacement."

exit 1

fi

OUTPUT:-

Administrator DESKTOP-TIC5DM4 MINGW64/ (master)

$ echo "Replacement completed. Modified text saved to 'modified.txt"." Replacement completed.

Modified text saved to 'modified.txt'.

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

LOGIC(CODE):-

$ vim wipro.sh

$ cat wipro.sh

#!/bin/bash

LOG_FILE="wipro.log"

grep "ERROR" "$LOG_FILE" | awk '{print $1,$2.$3,$4,$5,$6,$7,$8}'

OUTPUT:-

Administrator@DESKTOP-TIC5DM4 MINGW64 ~/git_demo2 (master)

$ nano wipro.sh

$ chmod +x wipro.sh

$ ./wipro.sh

2024-06-25 10:17:45ERROR Failed to load configuration

2024-06-25 10:25:00ERROR Unable to connect to database

Assignment 7:

Create a script that takes a text file and replaces all occurences of “old_text” with

“new text”.use sed to perform this operation and output the result of new life.

LOGIC(CODE):-

$#!/bin/bash

Administrator@DESKTOP-TIC50M4 MINGW64/(master)

5 text="This is some text with old_text in it. Here's another occurrence of old_text."

$ echo "Stext"> original.txt

$ old_text="old_text"

new_text="new_text"

Administrator DESKTOP-TIC50M4 MINGW64/ (master)

$ sed "s/Sold_text/$new_text/g" original.txt > modified.txt

Administrator@DESKTOP-TICSDM4 MINGW64/ (master)

$ echo "Replacement completed. Modified text saved to 'modified.txt'." Replacement completed.

Modified text saved to 'modified.txt'.

OUTPUT:-

$ echo "Replacement completed. Modified text saved to 'modified.txt'." Replacement completed.

Modified text saved to 'modified.txt'.