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: first I have created a file named "myfile.txt"

```
#i/bin/bash
filename="myfile.txt"

if [ -e "$filename" ];then
  echo "File exists"
else
   echo "File not found"
fi
~
```

Output : File exists

Assignment2. 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.

C:\Program Files\WSL\wsl.exe

```
#i/bin/bash

while true; do
read -p "Enter a number (0 to stop): " num
if [ "$num" -eq 0 ]; then
break
elif [ $((num % 2)) -eq 0 ]; then
  echo "$num is even"
else
   echo "$num is odd"
fi
done
~
```

Output:

```
DESKTOP-TIC5DM4:~/wipro/myscriptss# ./ss6.sh
Enter a number (0 to stop): 5
5 is odd
Enter a number (0 to stop): 4
4 is even
Enter a number (0 to stop): 2
2 is even
Enter a number (0 to stop): 77
77 is odd
Enter a number (0 to stop): 67
67 is odd
Enter a number (0 to stop): 34
34 is even
Enter a number (0 to stop): 23
23 is odd
Enter a number (0 to stop): 0
DESKTOP-TIC5DM4:~/wipro/myscriptss#
```

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: I have created two files : myfile.txt and ss6.sh.

```
DESKTOP-TIC5DM4:~/wipro/myscriptss# ls
TestDir          ss1.sh          ss4.sh          ss7.sh
myfile.txt       ss2.sh          ss5.sh          ss8.sh
shelldcript1.sh  ss3.sh          ss6.sh
DESKTOP-TIC5DM4:~/wipro/myscriptss# cat myfile.txt
this i new file
DESKTOP-TIC5DM4:~/wipro/myscriptss# cat ss6.sh
#i/bin/bash
this is another file
i have edited the code
a b c
d e f
g h i
j k l
DESKTOP-TIC5DM4:~/wipro/myscriptss#
```

Script:

C:\Program Files\WSL\wsl.exe

```
#i/bin/bash

function count_lines(){
local filename=$1
local number_of_lines=$(wc -l < "$filename")
echo "The file $filename has $number_of_lines lines."
}

count_lines "myfile.txt"
count_lines "ss6.sh"
```

Output:

```
DESKTOP-TIC5DM4:~/wipro/myscriptss# ./ss7.sh
The file myfile.txt has 1 lines.
The file ss6.sh has 7 lines.
DESKTOP-TIC5DM4:~/wipro/myscriptss#
```

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 filename as its content(e.g.,File1.txt contains "File1.txt").

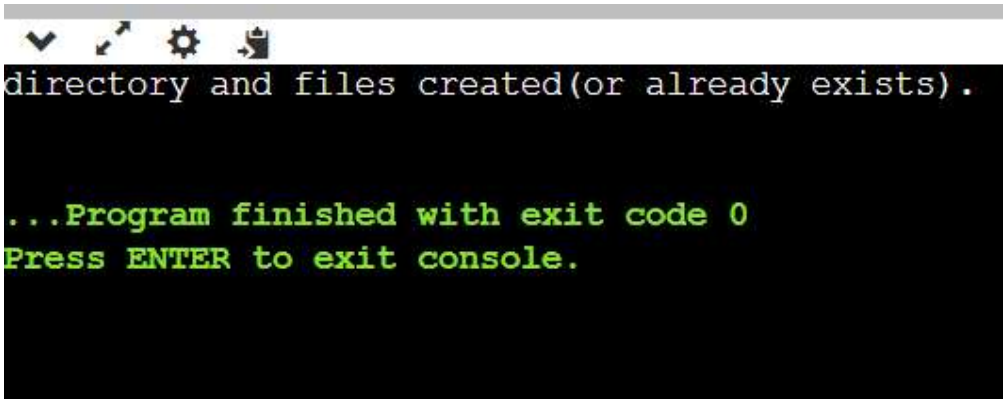
```
#                              Online Bash Shell.
#                 Code, Compile, Run and Debug Bash script online
# Write your code in this editor and press "Run" button to execut

dir_name="TestDir"

mkdir -p "$dir_name"

for i in {1..10}; do
  filename="File$i.txt"
  filepath="$dir_name/$filename"
  echo "$filename" > "$filepath"
done

echo "Directory '$dir_name' and files created successfully!"
```

This script creates a directory named "TestDir" and creates 10 files(File1.txt, File2.txt…File10.txt) and contains file names.

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.

```
Debug_mode=true

create_dir_with_handling() {
    local dir_name="$1";
    if [ "$debug_mode" = "true" ]; then
    echo "Creating directory: ' $dir_name' "
    fi
    mkdir -p "$dir_name" || {  echo "Error creating directory $dir_name' "; exit 1;}

}

create_dir_with_handling "TestDir"

for i in {1..10}; do
 filename="file$i.txt"

 if [ "$debug_mode" = "true" ]; then
   echo "creating file: '$filename' "
   fi
   echo "$filename" > "TestDir/$filename" || {  echo "Error creating file '$filename' ";  exit 1; }
   done
   echo "directory and files created(or already exists)."
```

Output:



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: firstly I have created a sample log file i.e logfile.log and entered some error message with date and time.

```
#i/bin/bash
2024-05-18 09:00:01 [INFO] System booting up
2024-05-18 09:15:23 [DEBUG] Checking configuration settings
2024-05-18 09:30:45 [ERROR] Unable to load configuration file
2024-05-18 10:00:30 [INFO] Configuration file loaded successfully
2024-05-18 10:15:54 [WARN] Disk space running low
2024-05-18 10:30:05 [ERROR] Failed to initialize network interface
2024-05-18 10:45:37 [ERROR] Network interface initialized
2024-05-18 11:00:00 [DEBUG] Service xyz state : running
2024-05-18 11:15:20 [ERROR] User login Attempt
~
~
```

Given this log file, we can use the following script to extract and format the lines containing "ERROR":

```
Select C:\Program Files\WSL\wsl.exe
#i/bin/bash
LOG_FILE="logfile.log"

grep "ERROR" "$LOG_FILE" | awk '{print $1, $2, substr($0, index($0, $5))}'
~
```

grep "ERROR" "$LOG_FILE": This extracts all lines containing the word "ERROR".
awk '{print $1, $2, substr($0, index($0,$5))}': This processes each extracted line to print the date, time, and the error message. Specifically:
$1: The date part (e.g., 2024-05-18).
$2: The time part (e.g., 10:25:33).
substr($0, index($0,$5)): This extracts the substring starting from the fifth field onward (which includes the error message.

Output:

```
DESKTOP-TIC5DM4:~/wipro/myscriptss# ./exract_error.sh
2024-05-18 09:30:45 to load configuration file
2024-05-18 10:30:05 to initialize network interface
2024-05-18 10:45:37 interface initialized
2024-05-18 11:15:20 login Attempt
DESKTOP-TIC5DM4:~/wipro/myscriptss#
```

The script correctly extracts the lines containing "ERROR" and prints the date, time, and the error message for each line.

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 :

At first we have to create two files, one is the original file for storing the content and the other one is the modified file.

Then we need to create a replace.sh file for writing the script

Original.txt content

```
DESKTOP-TIC5DM4:~/wipro/myscriptss# cat original.txt
This is some sample text file that contains old_text.
Here's another line with old_text appearing twice.
old_text can be at the begining of the line as well.
there can be multiple line without old_text too.
this is the last line and this also has the old_text.
DESKTOP-TIC5DM4:~/wipro/myscriptss#
```

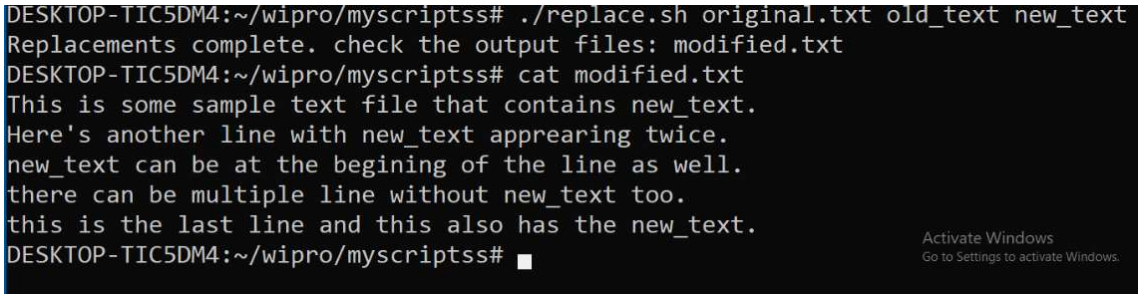
Script:

```
C:\Program Files\WSL\wsl.exe
#i/bin/bash
if [ "$#" -ne 3 ]; then
    echo "Usage: $0 input_file old_text new_text"
    exit 1
fi

input_file=$1
old_text=$2
new_text=$3
output_file="modified.txt"

sed "s/$old_text/$new_text/g" "$input_file" > "$output_file"
echo "Replacements complete. check the output files: $output_file"
```

And then we execute the script:

```
DESKTOP-TIC5DM4:~/wipro/myscriptss# ./replace.sh original.txt old_text new_text
```

After replacement of the text:


DESKTOP-TIC5DM4:~/wipro/myscriptss# ./replace.sh original.txt old_text new_text
Replacements complete. check the output files: modified.txt
DESKTOP-TIC5DM4:~/wipro/myscriptss# cat modified.txt
This is some sample text file that contains new_text.
Here's another line with new_text apprearing twice.
new_text can be at the begining of the line as well.
there can be multiple line without new_text too.
this is the last line and this also has the new_text.
DESKTOP-TIC5DM4:~/wipro/myscriptss#
Activate Windows
Go to Settings to activate Windows.