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".

```
#!/bin/bash

file_name="myfile.txt"
if [ -f "$file_name" ]; then
        echo "File exist"
else
        echo "File not exits"
fi

OUTPUT:

[root@localhost Kanchan]# chmod u+x pro1.sh
[root@localhost Kanchan]# bash pro1.sh
 File exist
```

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.

```
#!/bin/bash
while true;
do read -p "Enter a number (or '0' to quit): " user_input
if [[ "$user_input" == "0" ]];
then
echo "Exiting..."
break
fi
if [[ ! "$user_input" =~ ^[-+]?[0-9]+$ ]];
then
echo "Invalid input. Please enter a number."
continue
fi
number=$((user_input % 2))
if [[ $number -eq 0 ]];
then
echo "$user_input is even."
else echo "$user_input is odd."
fi
done
```

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.

```
#!/bin/bash

count_lines() {
  local filename="$1"
  if [[ -f "$filename" ]]; then
    line_count=$(wc -l < "$filename")
    echo "File: $filename - Line count: $line_count"
  else
    echo "Error: File '$filename' does not exist."
  fi
}

count_lines "myfile.txt"
count_lines "hello.txt"
count_lines "nonexistentfile.txt"

echo "All file checks completed."
```

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").

```
#!/bin/bash

dir_name="TestDir"
num_files=10

if [[ -d "$dir_name" ]]; then
  echo "Directory '$dir_name' already exists. Skipping directory
creation."
else
  mkdir -p "$dir_name"
  if [[ $? -eq 0 ]]; then
    echo "Directory '$dir_name' created successfully."
  else
    echo "Error: Failed to create directory '$dir_name'."
    exit 1
  fi
fi
for i in $(seq 1 $num_files); do
```

```
  file_name="File$i.txt"
  file_path="$dir_name/$file_name"
  echo "$file_name" > "$file_path"
  if [[ $? -eq 0 ]]; then
    echo "File '$file_path' created successfully."
  else
    echo "Error: Failed to create file '$file_path'."
  fi
done

echo "All files created successfully."
```

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.

```
#!/bin/bash

debug_mode=false

dir_name="TestDir"

num_files=10

print_debug_message() {
  if [[ $debug_mode == true ]]; then
    echo "DEBUG: $1"
  fi
}
if [[ -d "$dir_name" ]]; then
  print_debug_message "Directory '$dir_name' already exists."
  echo "Directory '$dir_name' already exists. Skipping creation."
else
  mkdir -p "$dir_name"
  if [[ $? -eq 0 ]]; then
    print_debug_message "Directory '$dir_name' created successful
ly."
    echo "Directory '$dir_name' created successfully."
  else
    echo "Error: Failed to create directory '$dir_name' (check pe
rmissions)."
    exit 1
  fi
fi

for i in $(seq 1 $num_files); do
  file_name="File$i.txt"
  file_path="$dir_name/$file_name"
```

```
  echo "$file_name" > "$file_path"
  if [[ $? -eq 0 ]]; then
    print_debug_message "File '$file_path' created successfully."
    echo "File '$file_path' created successfully."
  else
    echo "Error: Failed to create file '$file_path' (check permis
sions)."
  fi
done
```

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

```
#!/bin/bash

log_file="sample.log"

error_lines=$(grep "ERROR" "$log_file")

for line in $error_lines; do
  date_time=$(echo "$line" | awk '{print $1,$2}')
  error_msg=$(echo "$line" | awk '{print $3}')

  echo "Date/Time: $date_time - Error Message: $error_msg"
done
echo "Finished processing log file '$log_file'."
```

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.

```
#!/bin/bash


if [ $# -ne 3 ]; then
  echo "Usage: $0 <input_file> <old_text> <new_text>"
  echo "  - input_file: The file to process."
  echo "  - old_text: The text to be replaced."
  echo "  - new_text: The replacement text."
```

```
  exit 1
fi


input_file="$1"
old_text="$2"
new_text="$3"

output_file="${input_file}_modified.txt"  # Modify as needed


if [ ! -f "$input_file" ]; then
  echo "Error: Input file '$input_file' does not exist."
  exit 1
fi

sed -i "s/$old_text/$new_text/g" "$input_file"

echo "Replacements completed. Modified content written to '$input
_file'."
```