Assignment:5 Modify the shell 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.

- Handling Errors: Check if the directory already exists before attempting to create it. If it exists, you can either abort the script or prompt the user for further action. Check for permissions to create files in the specified directory. If the script lacks the necessary permissions, it should gracefully handle the error, informing the user and possibly exiting with error code.

- Debugging Mode: Add an option to enable debugging mode, which will print additional information during script execution. This can include variable values, intermediate steps, or any other information that helps in understanding the script's behavior

```
#!/bin/bash

debug() {
   if [ "$DEBUG_MODE" = true ]; then
      echo "DEBUG: $1"
   fi
}

handle_error() {
   echo "Error: $1"
   exit 1
}

DEBUG_MODE=false
if [ "$1" = "-d" ]; then
   DEBUG_MODE=true
   debug "Debug mode enabled."
fi

if [ -d "TestDir" ]; then
   handle_error "Directory 'TestDir' already exists."
fi
mkdir -p TestDir || handle_error "Failed to create directory 'TestDir'. Check permissions."

for ((i=1; i<=10; i++)); do

   echo "File$i.txt" > TestDir/File$i.txt || handle_error "Failed to create file 'File$i.txt'.
Check permissions."
   debug "File 'File$i.txt' created successfully."
```

```
done

echo "Files created successfully in TestDir."
```

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.

```
#!/bin/bash

# Check if a filename is provided as an argument
if [ $# -ne 1 ]; then
   echo "Usage: $0 <log_file>"
   exit 1
fi

log_file="$1"

# Check if the log file exists
if [ ! -f "$log_file" ]; then
   echo "Error: Log file '$log_file' not found."
   exit 1
fi

# Use grep to extract lines containing "ERROR" and awk to print date, time, and error message
grep "ERROR" "$log_file" | awk '{print $1, $2, $0}'
```

This script will extract all lines containing "ERROR" from the provided log file and print the date, time, and error message of each extracted 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.

```
#!/bin/bash

# Check if two arguments are provided
```

```
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"

# Check if the input file exists
if [ ! -f "$input_file" ]; then
    echo "Error: Input file '$input_file' not found."
    exit 1
fi

# Perform the replacement using sed and overwrite the original file
sed -i "s/$old_text/$new_text/g" "$input_file"
echo "Replacement complete. Check '$input_file' for updated content."
```

This script will replace all occurrences of "old_text" with "new_text" in the specified input file using `sed` and overwrite the original file with the updated content.