**Shell Scripting**

**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".**

filename="myfile.txt"

if [ -f "$filename" ]; then

echo "File exists"

else

echo "File not found"

fi

**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.**

check\_odd\_even() {

if [ $(($1 % 2)) -eq 0 ]; then

echo "The number $1 is even."

else

echo "The number $1 is odd."

fi

}

while true; do

echo "Enter a number (0 to quit):"

read number

if [ "$number" -eq 0 ]; then

echo "Exiting..."

break

fi

check\_odd\_even $number

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.**

count\_lines() {

local filename="$1"

if [ -f "$filename" ]; then

local line\_count=$(wc -l < "$filename")

echo "The file '$filename' has $line\_count lines."

else

echo "The file '$filename' does not exist."

fi

}

count\_lines "file1.txt"

count\_lines "file2.txt"

count\_lines "myfile.txt"

**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").**

mkdir -p TestDir

cd TestDir

for i in {1..10}; do

filename="File$i.txt"

echo "$filename" > "$filename"

done

echo "Created 10 files in TestDir with their filenames as content."

**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=false

debug() {

if [ "$DEBUG" = true ]; then

echo "DEBUG: $1"

fi

}

dir\_name="TestDir"

debug "Attempting to create directory $dir\_name"

if mkdir -p "$dir\_name"; then

echo "Directory $dir\_name created successfully."

else

echo "Failed to create directory $dir\_name." >&2

exit 1

fi

debug "Changing to directory $dir\_name"

if cd "$dir\_name"; then

debug "Changed to directory $dir\_name"

else

echo "Failed to change to directory $dir\_name." >&2

exit 1

fi

for i in {1..10}; do

filename="File$i.txt"

debug "Creating file $filename with its filename as content"

if echo "$filename" > "$filename"; then

debug "File $filename created successfully"

else

echo "Failed to create file $filename." >&2

exit 1

fi

done

echo "Created 10 files in $dir\_name with their filenames as content."

**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**

log\_file="sample.log"

if [ ! -f "$log\_file" ]; then

echo "Log file $log\_file does not exist." >&2

exit 1

fi

grep "ERROR" "$log\_file" | awk '{print $1, $2, substr($0, index($0,$4))}'

sed 's/ERROR/ALERT/g' "$log\_file" > processed\_log.log

echo "Processed log file has been saved to processed\_log.log with 'ERROR' replaced by 'ALERT'."

**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.**

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="output\_$(basename "$input\_file")"

if [ ! -f "$input\_file" ]; then

echo "Input file $input\_file does not exist." >&2

exit 1

fi

sed "s/$old\_text/$new\_text/g" "$input\_file" > "$output\_file"

echo "Replaced all occurrences of '$old\_text' with '$new\_text' in $input\_file."

echo "Output saved to $output\_file."