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".

Solution

```
#!/bin/bash

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.

Solution

```
#!/bin/bash

while true;
do
    echo "Enter a number "
    read number
```

```
    # Check if the input is 0 the program ends
    if [ $number -eq 0 ]
    then
        echo "0 found"
        break
    fi


    if [ $((number % 2)) -eq 0 ]
    then
        echo "$number is even"
    else
        echo "$number 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.

Solution

```
#!/bin/bash

# Define the function
count_lines() {
    filename="$1"
    if [ -f "$filename" ];
    then
        lines=$(wc -l < "$filename")
        echo "Number of lines in $filename: $lines"
    else
        echo "$filename does not exist or is not a regular file"
```

```
    fi
}

# Call the function with different filenames
count_lines "file1.txt"
count_lines "file2.txt"
count_lines "file3.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").

Solution

```
#!/bin/bash

# Create the directory TestDir
mkdir -p TestDir

# Move into the directory
cd TestDir || exit

# Loop to create ten files
i=1
while [ "$i" -le 10 ]
do
    filename="File${i}.txt"
    content="$filename"
    echo "$content" > "$filename"
    i=$((i + 1))
done

echo "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.

Solution

```
#!/bin/bash

set -x
directory="Sudheer"

if [ -d "$directory" ]
then
    echo "Directory  exists."
else
    mkdir -p "$directory"
    echo "Directory created."
fi
set +x
```

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

Solution

```
#!/bin/bash
```

```
# Sample log file path
logfile="sample.log"

# Use grep to extract lines containing "ERROR" and pass it to awk
for processing
grep "ERROR" "$logfile" | \
awk '{
    # Extract date and time
    date_time = $1 " " $2

    # Remove date and time from the original line
    $1=$2=""

    # Print date, time, and the rest of the line (error message)
    print date_time, $0
}'
```

sample.log

```
----------
2024-05-16 08:30:15 INFO: Application started
2024-05-16 08:31:22 ERROR: Database connection failed
2024-05-16 08:32:45 WARNING: Disk space low
2024-05-16 08:33:12 ERROR: Invalid input received
2024-05-16 08:34:55 ERROR: Server crashed
```

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.

Solution
---

replace-sed.sh

```
#!/bin/bash


input_file="input.txt"

sed 's/red/blue/g' "$input_file" > "output.txt"

echo "Text replaced successfully. Output written to 'output.txt'."
```

input.txt

```
red red red red red red red
red red
red red red red yellow
```