Manjula Nannuri

Day-7 Assignment

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
```

OUTPUT:

```
[root@localhost ~]# bash myfile.txt
file exists
```

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.

```
while true; do
        echo -n " enter a nmber (0 to quit):"
        read n

        if [ "$n" -eq 0 ]; then
                echo "Exiting..."
                break
        fi

        if [ $(expr $n % 2) -eq 0 ]; then
                echo "$n is an even number"
        else
                echo "$n is odd number"
        fi
done
```

OUTPUT:

```
"oddeven.sh" [New] 16L, 229B written
[root@localhost ~]# chmod u+x oddeven.sh
[root@localhost ~]# bash oddeven.sh
 enter a nmber (0 to quit):4
4 is an even number
 enter a nmber (0 to quit):5
5 is odd number
```

```
 enter a nmber (0 to quit):0
Exiting...
```

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"
count_lines() {
local filename="$1"

local line_count=0
#line_count=$(wc -1 < "$filename")

echo "the file '$filename' has '$line_count' lines (simulated)."

}

filenames = "abc.txt" "sample.log"
for filename in "{filenames[@]}"; do
count_lines "$filename"
done

echo "script execution complte"
```

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").

```
directory="TestDir"

if [ ! -d "$directory" ]; then

        mkdir "$directory"

        echo "Directory '$directory' created."

else

        echo "Directory '$directory' already exists."

fi


for i in {1..10}; do
```

```
        filename="File$i.txt"
        filepath="$directory/$filename"
        echo "$filename" > "$filepath"
        echo "created file '$filepath' with content '$filename
'."
done
```

OUTPUT:

```
"diff_filenames.sh" 15L, 341B written
[root@localhost ~]# bash diff_filenames.sh
Directory 'TestDir' already exists.
created file 'TestDir/File1.txt' with content 'File1.txt'.
created file 'TestDir/File2.txt' with content 'File2.txt'.
created file 'TestDir/File3.txt' with content 'File3.txt'.
created file 'TestDir/File4.txt' with content 'File4.txt'.
created file 'TestDir/File5.txt' with content 'File5.txt'.
created file 'TestDir/File6.txt' with content 'File6.txt'.
created file 'TestDir/File7.txt' with content 'File7.txt'.
created file 'TestDir/File8.txt' with content 'File8.txt'.
created file 'TestDir/File9.txt' with content 'File9.txt'.
created file 'TestDir/File10.txt' with content 'File10.txt'.
```

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.

```
create_directory() {
if [ ! -d "&1" ]; then
        mkdir -p "$1"
        echo "Directory '$1' created successfully."
else
        echo "Directory '$1' already exists."
fi
}

main() {
directory=$1
debug_mode=$2

if [ "$debug_mode" = "--debug" ]; then
        echo "Debug mode enabled."
fi
create_directory "$directory"
}

if [ "$#" -eq 0 ]; then
        echo "Usage: $0 <directory_name> [--debug]"
exist 1
fi
main "$@"
~
~
~
```

OUTPUT:

```
"create_directory.sh" 24L, 384B written
[root@localhost ~]# ./create_directory.sh new_directory --debug
Debug mode enabled.
Directory 'new_directory' created successfully.
[root@localhost ~]#
```

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.

```
'grep_log.sh" [New] 7L, 321B written
[root@localhost ~]# chmod +x grep_log.sh
[root@localhost ~]# ./grep_log.sh
sampe log file created at: sample.log
[root@localhost ~]# grep "ERROR" sample.log | awk '{print $1, $2, $0}'
2024-05-19 08:31:00 2024-05-19 08:31:00 ERROR: an error occured: File not fount
2024-05-19 08:32:00 2024-05-19 08:32:00 ERROR: another error occured: permission denied
[root@localhost ~]# cat sample.log
2024-05-19 08:30:00 INFO: This is an informational message
2024-05-19 08:31:00 ERROR: an error occured: File not fount
2024-05-19 08:32:00 ERROR: another error occured: permission denied
[root@localhost ~]#
```

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 sample.log occured encounted "
        exit 1
fi

input_file="$1"

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

old_text ="$2"
new_text ="$3"

out_file="${input_file%.text}_modified.text"

sed "s/$old_text/$new_text/g" "$input_file" > "$output_file"

echo "text replacement completed. Result saved to $output_file."
~
~
```

OUTPUT:

```
abc_modified.txt" [New] 0L, 0B written
root@localhost ~]# cat abc_modified.txt
root@localhost ~]# ./replaceScript.sh abc.txt ram laxman
```