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".

Answer: filename="myfile.txt"

#!/bin/bash

if [-e "$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.

```
#!/bin/bash

while true; do
    # Read a number from the user
    echo "Please enter a number (0 to quit):"
    read number

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

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

```
venkatesh@venky:~$ vim ss1.sh
venkatesh@venky:~$ chmod 777 ss4.sh
chmod: cannot access 'ss4.sh': No such file or directory
venkatesh@venky:~$ chmod 777 ss1.sh
venkatesh@venky:~$ ./ss1.sh
./ss1.sh: line 12: syntax error near unexpected token `('
./ss1.sh: line 12: `        if [ $ ((number%2)) -eq 0 ]; then'
venkatesh@venky:~$ vim ss1.sh
venkatesh@venky:~$ ./ss1.sh
please enter a number ( 0 to Quit);
6
6 is even.
please enter a number ( 0 to Quit);
10
10 is even.
please enter a number ( 0 to Quit);
3
3 is odd.
please enter a number ( 0 to Quit);
0
Exiting...
venkatesh@venky:~$
```

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.

main.bash | f.txt | f1.txt | f3.txt

```bash
linecount(){
        file="$1"
        lines=$(wc -l < "$file")
        echo "no.of lines in file:$lines"

}
linecount f.txt
linecount f1.txt
linecount f3.txt
```

```
no.of lines in file:4
no.of lines in file:3
no.of lines in file:1


...Program finished with exit code 0
Press ENTER to exit console.
```

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"").

main.bash | Testdir/File1.txt | Testdir/File10.txt | Testdir/File2.txt | Testdir/File3.txt | Testdir/File4.txt | Testdir/File5.txt | Testdir/File6.txt | Testdir/File7.txt | Testdir/File8.txt

```bash
mkdir Testdir
cd Testdir
for ((i=1; i<=10; i++));
do
    filename="File$i.txt"
    echo $filename > $filename
done
```

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.

main.bash | TestDir/File1.txt | TestDir/File10.txt | TestDir/File2.txt | TestDir/File3.txt | TestDir/File4.txt | TestDir/File5.txt | TestDir/File6.txt | TestDir/File7.txt | TestDir/File8.txt

```bash
DEBUG=false
if [ "$DEBUG" = "true" ]; then
    set -x
fi
# Function to print debug messages
debug_msg() {
    if [ "$DEBUG" = "true" ]; then
        echo "[DEBUG] $1"
    fi
}
# Create a directory named TestDir
debug_msg "Creating directory TestDir"
if ! mkdir -p TestDir 2>/dev/null; then
    echo "Error: Could not create directory TestDir"
    exit 1
fi

# Change the current working directory to TestDir
debug_msg "Changing directory to TestDir"
if ! cd TestDir 2>/dev/null; then
    echo "Error: Could not change to directory TestDir"
    exit 1
fi

# Create ten files named File1.txt, File2.txt, ..., File10.txt
for i in {1..10}
do
    filename="File$i.txt"
    debug_msg "Creating file $filename with content $filename"
    if ! echo $filename > $filename; then
        echo "Error: Could not create file $filename"
        exit 1
    fi
done
debug_msg "Script  completed successfully"
```

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

main.bash | sample.log

```bash
logfile="sample.log"

grep "ERROR" "$logfile" | awk '{print $1, $2, $NF}'
```

```
2024-05-19 12:35:01 application
2024-05-19 12:35:15 occurred


...Program finished with exit code 0
Press ENTER to exit console.
```

main.bash | sample.log

```
2024-05-19 12:34:56 INFO Some informational message
2024-05-19 12:35:01 ERROR An error occurred in the application
2024-05-19 12:35:10 WARNING A warning message
2024-05-19 12:35:15 ERROR Another error occurred
```

```
2024-05-19 12:35:01 application
2024-05-19 12:35:15 occurred


...Program finished with exit code 0
Press ENTER to exit console.
```

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.

main.bash | file1.txt | file2.txt

```bash
input="$1"
old="$2"
new="$3"
out="output_$(basename "$input")"
if [ ! -f "$input" ]; then
    echo "Error: File not found!"
    exit 1
fi
sed "s/$old/$new/g" "$input" > "$out"
```

```
Error: File not found!


...Program finished with exit code 1
Press ENTER to exit console.
```