**Assignment 1: Ensure the script checks if a specific file (e.g.. myfile.txt) exists in the current directory. If it costs, print "File exists", otherwise print "File not found".linux**

```
#!/bin/bash


if [ -e "myfile.txt" ]; 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
   echo "Enter a number (0 to exit): "
   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
```

**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.**

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

```
count_lines() {
   local filename="$1"
   local num_lines=$(wc -l < "$filename")
   echo "Number of lines in $filename: $num_lines"
}

count_lines "file1.txt"
count_lines "file2.txt"
```

**Assignment 4: Write a script that creates a directory named Test Dir and inside it, creates ten files named Filel.txt. File2.txt File10.txt. Each file should contain its filename as its content (e.g. Filel .txt contains "Filel.txt").**

```
#!/bin/bash

mkdir "Test Dir"

cd "Test Dir" || exit

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

**Assignment 5: Modify the script to handle crrors, such as the directory already existing or lacking permissions to create files Add a debugging mode that prints additional information when enabled.**

```
#!/bin/bash

handle_error() {
   local error_message="$1"
   echo "Error: $error_message"
```

```
    exit 1
}

create_files() {
    local directory_name="Test Dir"
    local debug_mode="$1"

    if [ -d "$directory_name" ]; then
        handle_error "Directory already exists."
    fi

    mkdir "$directory_name" || handle_error "Failed to create directory."



    cd "$directory_name" || handle_error "Failed to change directory."

    for ((i=1; i<=10; i++)); do
        filename="File$i.txt"
        echo "$filename" > "$filename" || handle_error "Failed to create file $filename."
        if [ "$debug_mode" = "true" ]; then
            echo "Created file: $filename"
        fi
    done

    echo "Ten files created with filenames File1.txt to File10.txt and contents set accordingly."
}
if [ "$1" = "--debug" ]; then
    create_files "true"
else
    create_files "false"
fi
```

**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**

```
divide(){
   numerator=$1
   denominator=$2

   result=$((numerator/denominator))
   echo "Result: $result"
}
handlerforError(){
    echo "error:dont do division by 0"
   exit 1
}
trap "handelerforErrot()" ERROR
trap "echo Hello World" SIGINT

divide 10 2
divide 2 0
#!/bin/bash

LOG_FILE="logfile.txt"

grep "ERROR" "$LOG_FILE" | \
awk '{date_time = $1 " " $2  $1 = ""; $2 = ""; print date_time, $0}'
```

**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.**

```
#!/bin/bash

if [ "$#" -ne 3 ]; then
    echo "Usage: $0 <input_file> <old_text> <new_text>"
    exit 1
fi

input_file="$1"

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

old_text="$2"

new_text="$3"

output_file="${input_file%.txt}_replaced.txt"

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

echo "Replacement done. Result saved to '$output_file'."
```