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".

```
Administrator@DESKTOP-TIC5DM4 MINGW64 ~/Myproject (master)
$ touch Assignment1.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 ~/Myproject (master)
$ vim Assignment1.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 ~/Myproject (master)
$ cat Assignment1.sh

#!/bin/bash
#Take input from the user for the file name

echo "Enter the file name:"
read filename

#check if file is in current directory

if [ -e "$filename" ]; then
        echo "File exists"
else
        echo "File not found"
fi

Administrator@DESKTOP-TIC5DM4 MINGW64 ~/Myproject (master)
$ ls -l
total 5
-rw-r--r-- 1 Administrator 197121   0 May  9 11:27 100
-rw-r--r-- 1 Administrator 197121   0 May  9 12:16 Assignment1
```

-rwxr-xr-x 1 Administrator 197121 220 May 9 12:26 Assignment1.sh*

-rw-r--r-- 1 Administrator 197121 21 May 8 16:40 a.txt

-rw-r--r-- 1 Administrator 197121 52 May 8 17:03 code1.py

-rw-r--r-- 1 Administrator 197121 395 May 8 17:02 index.html

-rwxr-xr-x 1 Administrator 197121 134 May 9 11:28 script1.sh*

Administrator@DESKTOP-TIC5DM4 MINGW64 ~/Myproject (master)

$ ./Assignment1.sh

Enter the file name:

a.txt

File exists

Administrator@DESKTOP-TIC5DM4 MINGW64 ~/Myproject (master)

$ ./Assignment1.sh

Enter the file name:

c.txt

File not found

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.

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ touch Assignment2.sh

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ vim Assignment2.sh

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ cat Assignment2.sh

```
#!/bin/bash

while true; do
    # Prompt the user to enter a number
    echo "Enter a number (enter 0 to exit):"
    read number

    # Check if the input is '0', if so, exit the loop
```

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

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

```
Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ chmod u+x Assignment2.sh

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ ./Assignment2.sh
Enter a number (enter 0 to exit):
6
6 is even.
Enter a number (enter 0 to exit):
7
7 is odd.
Enter a number (enter 0 to exit):
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.

```
Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ touch Assignment3.sh

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ vim Assignment3.sh

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ cat Assignment3.sh
#!/bin/bash

# Define function to print number of lines in a file

line_count() {
    filename="$1"
    if [ -f "$filename" ]; then
        lines=$(wc -l < "$filename")
        echo "Number of lines in $filename: $lines"
    else
```

```
        echo "File $filename not found."
    fi
}

# Call the function with different filenames
line_count "file1.txt"
line_count "file2.txt"
line_count "file3.txt"
```

```
Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ ./Assignment3.sh
Number of lines in file1.txt: 3
Number of lines in file2.txt: 4
Number of lines in file3.txt: 4
```

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").

```
Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ touch Assignment4.sh

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ vim Assignment4.sh

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ cat Assignment4.sh
mkdir -p "directory1"
cd "directory1"
for ((i=1;i<=10;i++));
do
echo "File$i.txt" >File$i.txt
done

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ ./Assignment4.sh

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject
$ cd directory1/

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject/directory1
$ cat File9.txt
File9.txt

Durgabhavani@DESKTOP-TTU5E43 MINGW64 ~/Myproject/directory1
$ cat File5.txt
File5.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.

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ touch Assignment5.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ vim Assignment5.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ cat Assignment5.sh
#!/bin/bash

#Function that display debug messages

debug_msg() {
	if [ "$debug_mode" = "true" ]; then
		echo "debug: $1"
	fi
}

#Function to create directory and handle errors

create_dir() {
	if [ -d "$1" ]; then
		echo "Error: Directory '$1' already exists."
		exit 1
	fi

	debug_msg "Creating directory: $1"
	mkdir -p "$1" || {
		echo "Error: Unable to create directory '$1'."
		exit 1
	}
}

#Main script

debug_mode=false

#check if debuging mode enable

if [ "$1" = "--debug" ]; then
	debug_mode=true
	shift  #remove --debug from args
fi

#check if at least one arg is provided
```

```
if [ $# -eq 0 ]; then
        echo "Usage: $0 [--debug] directory_name"
        exit 1
fi

dir_name="$1"

#creating dir

create_dir "$dir_name"
echo "Directory '$dir_name' created successfully."
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ ./Assignment5.sh new_directory
Directory 'new_directory' created successfully.

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ ./Assignment5.sh Myproject
Directory 'Myproject' created successfully.

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ ./Assignment5.sh
Usage: ./Assignment5.sh enter directory_name
```

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

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ touch log_file.log


Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ vim log_file.log

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ cat log_file.log
echo "This is log file"
2024-05-09 08:30:20 - ERROR: Unable to connect to database
2024-05-09 07:45:12 - WARNING: Connection timeout
2024-05-09 09:36:38 - ERROR: Disk space full
2024-05-09 08:32:56 - INFO: Application started
2024-05-09 08:35:04 - ERROR: Server crashed unexpectedly

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ touch Assignment6.sh
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ vim Assignment6.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ cat Assignment6.sh
#!/bin/bash

#log file path

log_file="log_file.log"

#grep to extract error containing lines

error_lines=$(grep "ERROR" "$log_file")

#Iterate over each error line and use awk to print date,time and error msg

echo "$error_lines" | awk '{print $1, $2, $3, substr($0, index($0,$4))}'

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ ./Assignment6.sh
2024-05-09 08:30:20 - ERROR: Unable to connect to database
2024-05-09 09:36:38 - ERROR: Disk space full
2024-05-09 08:35:04 - ERROR: Server crashed unexpectedly
```

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.

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ touch file.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ vim file.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ cat file.txt
This is old file.

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ touch Assignment7.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ vim Assignment7.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ cat Assignment7.sh
```

```
#!/bin/bash

#Check 3 arguments are provided

if [ "$#" -ne 3 ]; then
        echo "Usage: $0 file old new"
        exit 1
fi

file="$1"
old="$2"
new="$3"
output="${file%.txt}_modified.txt" #append modified to original filename

#Check file exits
if [ ! -f "$file" ]; then
        echo "Error: Input file '$file' does not exist."
        exit 1
fi
#replace using sed

sed "s/$old/$new/g" "$file" > "$output"

echo "Replacement complete.Modified content saved to '$output'."


Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ ./Assignment7.sh file.txt old new
Replacement complete.Modified content saved to 'file_modified.txt'.

Administrator@DESKTOP-TIC5DM4 MINGW64 /Myproject (master)
$ cat file_modified.txt
This is new file.
```