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".

Run Debug Stop Share Save { } Beautify

main.bash myfile.txt ss1.sh

```bash
filename="ss1.sh"

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

```bash
while true; do
    # Prompt the user to enter a number
    echo -n "Enter a number (0 to quit): "
    read number

    # Check if the entered number is '0'
    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
```

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_in_file() {
    local filename="$1"
    if [ -f "$filename" ]; then
        local line_count=$(wc -l < "$filename")
        echo "$line_count"
    else
        echo "File not found"
        return 1
    fi
}

# Main script
for fname in "$@"; do
    line_count=$(count_lines_in_file "$fname")
    if [ $? -eq 0 ]; then
        echo "The file '$fname' has $line_count lines."
    else
        echo "The file '$fname' was not found."
    fi
```

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").

```
rps@rps-virtual-machine:~$ mkdir testdir
rps@rps-virtual-machine:~$ cd testdir
rps@rps-virtual-machine:~/testdir$ echo "File1.txt" > File1.txt
rps@rps-virtual-machine:~/testdir$ echo "File2.txt" > File2.txt
rps@rps-virtual-machine:~/testdir$ echo "File3.txt" > File3.txt
rps@rps-virtual-machine:~/testdir$ echo "File4.txt" > File4.txt
rps@rps-virtual-machine:~/testdir$ echo "File5.txt" > File5.txt
rps@rps-virtual-machine:~/testdir$ echo "File6.txt" > File6.txt
rps@rps-virtual-machine:~/testdir$ echo "File7.txt" > File7.txt
rps@rps-virtual-machine:~/testdir$ echo "File8.txt" > File8.txt
rps@rps-virtual-machine:~/testdir$ echo "File9.txt" > File9.txt
rps@rps-virtual-machine:~/testdir$ echo "File10.txt" > File10.txt
rps@rps-virtual-machine:~/testdir$ ls
File10.txt  File2.txt  File4.txt  File6.txt  File8.txt
File1.txt   File3.txt  File5.txt  File7.txt  File9.txt
rps@rps-virtual-machine:~/testdir$ cat file1.txt
cat: file1.txt: No such file or directory
rps@rps-virtual-machine:~/testdir$ cat File1.txt
File1.txt
rps@rps-virtual-machine:~/testdir$
```

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.

Run Debug Stop Share Save { } Beautify

main.bash myfile.txt ss1.sh

```bash
#! /bin/bash

set -x

echo -n "Enter a number"
read n


remainder=$(( $n % 2 ))

set +x

if [ $remainder -eq 0 ]
then
 echo "You have entered $n -- which is an Even number"
else
 echo "You have entered $n -- which is an Odd number"
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

```
.bash    myfile.txt    ss1.sh    sample.log
#!/bin/bash

# Define the log file
LOGFILE="sample.log"

# Use grep to extract lines containing "ERROR"
grep "ERROR" "$LOGFILE" | awk '
{
    # Assuming the log format is: YYYY-MM-DD HH:MM:SS [ERROR] Error message
    date=$1
    time=$2
    # Combine the rest of the line for the error message
    $1=$2=$3=""
    error_message=$0

    # Print date, time and error message
    print date, time, error_message
}' | sed 's/^\s*//;s/\s*$//'
```

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.

```
rps@rps-virtual-machine:~/testdir$ vim file1.txt
rps@rps-virtual-machine:~/testdir$ cat file1.txt

 hello world java
rps@rps-virtual-machine:~/testdir$ sed -i 's/hello world java/amit kumar/' file1.txt
rps@rps-virtual-machine:~/testdir$ cat file1.txt

 amit kumar
rps@rps-virtual-machine:~/testdir$
```