Shell Scripting with Bash

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 : vi checkfile.sh

#!/bin/bash

File = “myfile.txt”

If [ -e “$file” ]; then

Echo “file exit”

Else

Echo “file not found”

fi

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.

Filename: vi odd\_even.sh

While true; do

Echo –n “Enter a number (0 to quit):”

Read number

If [ “$number” –eq o ]; then

Echo “Exiting…”

Break

fi

if [ $((number % 2)) –eq 0 ];

then

echo “$number is even:

else

echo “$number is odd”

fi

done

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.

Filename: count\_lines

Count\_lines() {

Filename=”$1”

If [ -f “$filename” ]; then

Lines=$(wc -1 < $filename”)

Echo “Number of lines in $filename : $lines”

Else

Echo “$filename does exit”

Fi

}

Count\_lines”myfile1.txt”

Count\_lines”myfile2.txt”

OR

count\_lines() {

local filename=$1

local count=$(wc -l < "$filename")

echo "The file '$filename' has $count lines."

}

count\_lines "file1.txt"

count\_lines "file2.txt"

count\_lines "file3.txt

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").

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.