Day 7 Assignments

1.count the number of files and folder present in the directory. if possible take the directory path from user.

```
 #!/bin/bash

# Prompt user for directory path

echo "Please enter the directory path:"

read directory


# Check if the directory exists

if [ -d "$directory" ]; then

   # Count the number of files

   file_count=$(find "$directory" -type f | wc -l)

   # Count the number of directories

   dir_count=$(find "$directory" -type d | wc -l)


   echo "Number of files in the directory: $file_count"

   echo "Number of directories in the directory: $dir_count"

else

   echo "The directory does not exist."

Fi
```

Explanation:

1. Prompt User for Directory Path: The script prompts the user to enter the directory path and stores it in the variable directory.
2. Check if the Directory Exists: The script checks if the provided path is a valid directory using the -d option.
3. Count the Number of Files: The find command with -type f finds all files in the directory. The output is piped to wc -l to count the number of lines, which corresponds to the number of files.
4. Count the Number of Directories: The find command with -type d finds all directories in the directory. The output is piped to wc -l to count the number of lines, which corresponds to the number of directories.
5. Output the Counts: The script prints the number of files and directories found.

2.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".

```
#!/bin/bash

# Function to count files and directories in a given directory

count_files_and_directories() {

   local directory=$1

 # Check if the directory exists

   if [ -d "$directory" ]; then

      # Count the number of files

      local file_count=$(find "$directory" -type f | wc -l)

      # Count the number of directories

      local dir_count=$(find "$directory" -type d | wc -l)

echo "Number of files in the directory: $file_count"

      echo "Number of directories in the directory: $dir_count"

   else

      echo "The directory does not exist."

   fi

}

# Function to check if a specific file exists in the current directory

check_specific_file() {

   local file_name="myfile.txt"


   if [ -e "$file_name" ]; then

      echo "File exists"

   else
```

```
echo "File not found"
    fi
}
```

 # Main script execution

```
echo "Please enter the directory path:"
read directory
```

 # Call function to count files and directories

```
count_files_and_directories "$directory"
```

 # Call function to check for specific file

```
check_specific_file
```

Explanation:

1. Function to Count Files and Directories: The count_files_and_directories function takes a directory path as an argument, checks if the directory exists, counts the number of files and directories, and prints the counts.
2. Function to Check Specific File: The check_specific_file function checks if myfile.txt exists in the current directory using the -e option. It prints "File exists" if the file is found and "File not found" otherwise.
3. Main Script Execution:-

i)Prompts the user to enter the directory path and reads it into the directory variable. ii)Calls the count_files_and_directories function with the entered directory path.iii)Calls the check_specific_file function to check if myfile.txt exists in the current directory.

3.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

# Function to check if a number is odd or even

check_odd_or_even() {
    local number=$1
    if [ $((number % 2)) -eq 0 ]; then
```

```
        echo "$number is even"
    else
        echo "$number is odd"
    fi
}
# Main script execution
while true; do
    echo "Please enter a number (enter '0' to stop):"
    read number
 # Check if the entered number is '0'
    if [ "$number" -eq 0 ]; then
        echo "You entered 0. Exiting..."
        break
    fi
  # Call function to check if the number is odd or even
    check_odd_or_even "$number"
done
```

Explanation:

1. **Function to Check Odd or Even:** The check_odd_or_even function takes a number as an argument, checks if it is even or odd using the modulo operation (%), and prints the result.
2. **Main Script Execution:**
    - The script enters an infinite while loop that continuously prompts the user to enter a number.
    - It reads the entered number into the number variable.
    - If the entered number is '0', it prints a message and breaks out of the loop.
    - Otherwise, it calls the check_odd_or_even function to check if the number is odd or even.

4.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

# Function to print the number of lines in a file

    print_number_of_lines() {

       local filename=$1

 # Check if the file exists

       if [ -f "$filename" ]; then

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

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

       else

          echo "The file '$filename' does not exist."

       fi

    }

# Main script execution



    # Call the function with different filenames

    print_number_of_lines "file1.txt"

    print_number_of_lines "file2.txt"

    print_number_of_lines "file3.txt"
```

Explanation:

1. **Function to Print Number of Lines:** The print_number_of_lines function takes a filename as an argument, checks if the file exists, counts the number of lines using wc -l, and prints the result. If the file does not exist, it prints an error message.
2. **Main Script Execution:** The script calls the print_number_of_lines function with different filenames (file1.txt, file2.txt, and file3.txt). You can replace these filenames with the actual filenames you want to check.