Lab Exercises

1. Write scripts /commands / syntax to print a message “Welcome to the World of Shell Scripting” 10 times.

2. Write scripts /commands / syntax to take input and print your name, degree program information, batch No and course title.

3. Write scripts /commands / syntax that take input and check the number is positive or negative.

4. Write scripts /commands / syntax that take input and check the number is even, odd or zero.

5. Write scripts /commands / syntax that take input and check the number is prime or not.

6. Write scripts /commands / syntax that take input and check the maximum and minimum of 5 input numbers.

7. Write scripts /commands / syntax to generate the Fibonacci series.

8. Write scripts /commands / syntax to find out the factorial of given input number.

9. Write scripts /commands / syntax for moving files into three subdirectories directories:

   shelldir, cdir, jpgdir according to their extensions. (Create at least 6 files first 2 for each extension).

10. Write a script that contains the following options based on user`s choice:
    - It allows you to create folders and files (quantity and name should be user input).
    - It allows you to change rights of your files.
    - It can help you in searching files.
    - It allows you to display processes like a task manager in Windows and should allow to kill any selected process.
    - Allows to write and run a new script (main script should always be in running).
    - Allows to write and run a C language program (main script should always be in running).

Make necessary assumptions to design your script that implements all the required options (mention your assumptions).

11. File Renaming Utility: You are tasked with creating a bash script that renames multiple files in a directory according to a specified naming convention. The script should:
    - Accept two arguments: the directory path containing the files and the new file name pattern.
    - Rename each file in the directory by appending a sequential number to the new file name pattern (e.g., `file1.txt`, `file2.txt`, etc.).
    - Preserve the original file extension during the renaming process.
    - Provide feedback to the user about the renaming process, including any errors encountered.

12. Directory Cleanup Script: Develop a bash script to automate directory cleanup tasks by removing old files and directories. The script should:
    - Accept a directory path as a command line argument.
    - Identify and delete files older than a specified number of days.
    - Recursively remove empty directories within the specified directory.
    - Provide feedback to the user about the cleanup process, including the number of files and directories removed.