# Bash Scripting Tasks

**Note:** You need to create a github repository and push your scripts to that repository

**Time:** 2 days

**Q1:** Create a backup script that copies all files in a specific directory to another directory, with the option to compress the files into a tarball for efficient storage.

**Q2:** Write a script that searches for a specific keyword in a directory of text files, and prints the line numbers and filenames where the keyword is found. The result should be displayed on the terminal as well as stored in a file ‘output.txt’.

**Q3:** Create a script that automatically generates a report of system usage, including CPU and memory usage, disk space, and network statistics.

**Q4:** Write a script that monitors a log file in real-time and sends an alert when a certain event occurs, such as an error or warning message.

**Q5:** Develop a script that automatically updates a Git repository with new changes(stage & commit changes), and pushes those changes to a remote repository. This can be useful for automating the deployment process of a web application.

**Q6:** You have a CSV file ([link here](https://drive.google.com/file/d/1A7YRrfJDJa2-UK6XxVYpw1HlKt6jXUMj/view?usp=sharing)) with the following columns: "First Name", "Last Name", "Email", and "Phone Number". Your task is to create a script that will perform the following tasks:

1. Check that the CSV file exists and is readable.
2. Create a new file called "formatted.csv" in the same directory as the input CSV file.
3. For each row in the input CSV file, convert the phone number to the format "(xxx) xxx-xxxx". If the phone number is not in the correct format, skip the row and log a warning message to a separate file called "warnings.log".
4. Concatenate the first and last name columns and store the result in a new column called "Full Name".
5. Add a new column called "Email Domain" to the end of each row, which contains only the domain of the email address (e.g. "gmail.com").
6. Sort the rows of the CSV file by the "Full Name" column in alphabetical order.
7. Write the formatted CSV file to the "formatted.csv" file.
8. Create a summary report in the terminal that shows the following information:

* The total number of rows in the input file.
* The number of rows that were successfully formatted and written to the output file.
* The number of rows that were skipped due to invalid phone numbers.
* A list of the top 10 most common email domains in the input file.

The script should be able to handle errors gracefully and provide informative error messages if necessary.