Answer any “15” question from the following.

1. Display the following options with dynamic values (within <> brackets) whenever you open/invoke powershell session.
    - Logged in username: <username>
    - Internet connectivity: <Active/Inactive>
    - Hostname: <hostname>
    - Current IP address: <fetch IP address>
    - Total number of running processes: <processes>
2. Write a script to create a variable containing a list of five numbers. Use a loop to calculate the square of each number and display the results.
3. Display all the services within a HTML webpage with the following customization like:
    - Name (starts with “S”)
    - Status (equals to “running”) &
    - Start Type (Automatic)
4. Write a command to retrieve the list of all files in a specific folder, filter those larger than 1MB, and display their names and sizes. Sort the results by size in descending order.
5. Write a powershell script that ping to multiple websites (of your choice) by the names from a text file. Display all pingable websites in GREEN color and non-pingable in RED color.
6. Create a function that accepts a folder path as a parameter, checks if the folder exists, and creates it if it does not. Log each operation to a file named `FolderCheckLog.txt`.
7. Write a custom powershell script with the help of 5 functions as custom cmdlets and access them using individual cmdlet.
    - List current execution policy
    - List HDD size information (total & free space).
    - List total running processes.
    - Number of processors (using environmental variable only)
    - BIOS Information
8. Write a script to iterate through an array of file paths, check if each file exists, and log whether the file is found or missing to a file named `FileCheckLog.txt`.
9. Delete $profile is already present manually. Write a PowerShell script to test if $profile exists, create $profile programmatically in case if file is not present.
10. Use PowerShell to retrieve and display the names of all processes consuming more than 50MB of memory. Save the results to a CSV file named `HighMemoryProcesses.csv`.
11. Use “non-persistent remoting” approach to create following options on remote computer(s).
    - Create a directory in C:\ drive and name it “CISLTIM”.
    - Create a file in this folder with name “PowerShell.txt”
    - Write your name and PS ID in this file.
12. Combine loops and conditional statements to create a script that generates numbers from 1 to 20, checks if each number is even or odd, and writes the results to a file named `EvenOddResults.txt`.
13. List top 10 processes “**Display Names**” with high CPU utilization and store these display names in a text file.

14. Create a script to list all services in a "Stopped" state, restart those that contain "Update" in their names, and log the actions to a file named `ServiceActionsLog.txt`.
15. Write a function that accepts only keywords like *"LTIMindtree", "Bhubaneshwar", "Odisha"* only and print/display "**Hello <>**" message when called.
16. Write a script to create a folder named `Project` with three subfolders: `Scripts`, `Reports`, and `Logs`. Inside the `Logs` folder, create a text file named `LogFile.txt` with the content: "Log file created successfully."
17. Change the execution policy to "AllSigned". Create a digitally signed powershell script with a new digital certificate with name "LTIMBBSR01" and verify if its executing.
18. Use PowerShell to retrieve the top five processes consuming the most CPU on your system and save their names, IDs, and CPU usage to a CSV file named `TopProcesses.csv`.
19. Write a PowerShell script that accepts service name from user and check if its running or not. If its running, then script must ask to either stop the service or exit. Similarly, if it is stopped, script must ask to run the service and must perform as input given by user.
20. Develop a PowerShell script to extract and display only the names and statuses of services containing the word "Network" in their names. Save the results to a text file named `NetworkServices.txt`.